

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА АВТОМАТИКИ ТА УПРАВЛІННЯ В ТЕХНІЧНИХ СИСТЕМАХ

«На правах рукопису»
УДК _____

«До захисту допущено»

Завідувач кафедри

_____ (підпис) _____ (ініціали, прізвище)

“ ” _____ 20__ р.

Магістерська дисертація

зі спеціальності (спеціалізації) 126, Інформаційні системи та технології, Інтегровані інформаційні системи

на тему: Кореляційний синхронний приймач системи передачі даних

Виконав : студентка 2 курсу, групи ІА-72мп
(шифр групи)

Ваціліна Галина Василівна _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент кафедри АУТС, к.т.н, доцент Полторак В.П. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

КИЇВ — 2019 РОКУ

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	5
ВСТУП.....	6
1 ОГЛЯД РІШЕНЬ ТА ТЕХНОЛОГІЙ	8
1.1 Порівняння з реалізацією приймача на інших обчислювальних платформах.....	11
1.1.1 Приймачі на основі GPU	11
1.1.2 Приймачі на основі ПЛІС FPGA.....	12
1.1.3 Приймачі на основі ASIC	14
1.2 Програмне забезпечення для реалізації моделей	15
1.2.1 Бібліотека Numpy	15
1.2.2 Бібліотека Scipy	16
1.2.3 Багатопотокове виконання.....	16
1.3 Огляд існуючих рішень для моделювання приймачів	18
1.3.1 GNU Octave (Signal Processing).....	18
1.3.2 Scilab	19
1.3.3 Програмний застосунок для математичних моделювання Matlab/Simulink.....	21
2 ТЕОРЕТИЧНИЙ ОПИС РІШЕННЯ	23
2.1 Кореляція.....	23
2.1.1 Коефіцієнт кореляції Пірсона	24
2.1.2 Коефіцієнт рангової кореляції Спірмена.....	25
2.2 Синхронізація приймача.....	26
2.2.1 Асинхронна та синхронна передачі даних.....	27
2.2.2 Синхронізація з використанням опорних сигналів	28
2.2.3 Фізичне кодування та самосинхронізуючі коди.....	29
2.2.4 Символьна синхронізація	32
2.2.4.1 Розімкнуті синхронізатори	33
2.2.4.2 Замкнуті синхронізатори	33

2.2.5 Кадрова синхронізація	34
2.2.6 Мережева синхронізація	38
2.2.7 Синхронізація приймача в мережах Ethernet.....	40
2.2.7.1 Fast Ethernet.....	40
2.2.7.2 Gigabit Ethernet	41
3 МОДЕЛЮВАННЯ ТА РЕАЛІЗАЦІЯ	42
3.1 Математичне моделювання процесів синхронізації.....	42
3.1.1 Частотна та фазова синхронізація	43
3.2 Математичне моделювання кореляційних функцій.....	46
3.2.1 Автокореляційна функція	46
3.2.2 Взаємнокореляційна функція (ВКФ) або взаємна кореляція двох сигналів.....	54
3.3 Моделювання АКФ та ВКФ в середовищі MatLab/Simulink	59
3.4 Реалізація моделі засобами Python	65
3.5 Реалізація моделі приймача засобами Matlab.....	66
3.6 Синхронний приймач на основі DSP	68
4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ	73
4.1 Вступ.....	73
4.2 Маркетинговий аналіз стартап-проекту.....	73
4.2.1 Опис ідеї проекту	73
4.2.2 Аналіз потенційних техніко-економічних переваг	75
4.3 Технологічний аудит ідеї проекту.....	77
4.4 Аналіз ринкових можливостей запуску стартап-проекту	78
4.4.1 Аналіз попиту	78
4.4.2 Визначення потенційних груп клієнтів.....	79
4.4.3 Аналіз ринкового середовища.....	79
4.4.4 Аналіз пропозиції	81
4.4.5 Аналіз умов конкуренції в галузі.....	82
4.4.6 Перелік факторів конкурентоспроможності.....	83

4.4.7 Аналіз сильних та слабких сторін стартап-проекту	84
4.4.8 SWOT-аналіз	85
4.4.9 Альтернативи ринкової поведінки	86
4.5 Розроблення ринкової стратегії проекту	86
4.5.1 Опис цільових груп потенційних клієнтів	87
4.5.2 Формування базових стратегій розвитку	87
4.5.3 Вибір стратегії конкурентної поведінки	88
4.5.4 Розробка стратегії позиціонування	89
4.6 Розробка маркетингової програми стартап-проекту	90
4.6.1 Формування маркетингової концепції	90
4.6.2 Визначення цінових меж	91
4.6.3 Визначення оптимальної системи збуту	92
4.6.4 Розробка концепції маркетингових комунікацій	92
4.7 Висновки	93
ВИСНОВКИ	95
ПЕРЕЛІК ПОСИЛАНЬ	97

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

PSK (Phase-shift keying) –

ФАПЧ – фазове автопідналаштування частоти

TDMA (time-division multiple access) –

FDMA (frequency-division multiple access) –

ВСТУП

Актуальність даної магістерської дисертації обумовлена високим розвитком технологій та потребою у високій якості передачі даних, в умовах великої кількості завад та не постійної передачі даних.

Об'єктом дослідження є система-емулятор, яка здійснюється синхронізацію з боку приймача методом кореляційного аналізу.

Предметом дослідження є створення системи-емулятору у вигляді програмного стенду для дослідження, демонстрації та вивчення синхронізації сигналів методом кореляційного аналізу.

Метою даної магістерської дисертації є створення кореляційного приймача із синхронізацією для систем передавання даних. На даний момент не існує програмного забезпечення, яке б демонструвало роботу, подібного роду, приймачів, а також дослідження даної сфери обмежене в більшій мірі теоретичним описом та відсутністю практичних прикладів.

Для досягнення мети магістерської дисертації було поставлено наступні задачі:

- огляд та дослідження предметної області;
- огляд існуючих патентів, рішень в області синхронізації приймачів;
- дослідження практичних рішень бітової та кадрової синхронізації в мережах Ethernet;
- дослідження теоретичних засад алгоритмів кореляції та створення алгоритму синхронізації з використанням кореляційних функцій;
- створення прототипу програмного комплексу для підтвердження працездатності складених алгоритмів;
- розробка програмного комплексу, що включатиме в себе розроблені алгоритми синхронізації в рамках магістерської дисертації, а також набір готових рішень, які дозволять якісно оцінити різницю реалізацій.

В рамках магістерської дисертації дослідження предметної області, інструментів, проблем проводилось з використанням загальнонаукових та спеціальних методів, а саме:

- метод абстрагування – представлення сигналів у вигляді числових рядів та робота з алгоритмами з використанням простих числових послідовностей;
- метод індукції та дедукції – виявлення потенційних недоліків в алгоритмах (швидкість роботи та помилки) під час їх виведення та врахування потенційних рішень для вирішення швидкості роботи;
- метод порівняння – огляд схожих варіантів реалізації синхронізації, порівняння різних алгоритмів знаходження коефіцієнтів кореляції, що дало змогу краще дослідити предметну область та розібратись в технічних вимогах до програмного комплексу;
- метод моделювання – розробка концепції на ранніх стадіях створення системи;
- аналіз та синтез – проведення дослідження існуючих варіантів синхронізації, створення нових рішень та розробка програмного комплексу з використанням високотехнічних засобів.

Розробка програмного комплексу-емулятора та саме кореляційного приймача породила ряд проблем, які були виявлені на етапі використання з реальними даними, а саме:

- 1) вибір технології для розробки програмного комплексу – один з найважливіх кроків, який задасть “тон” та швидкість розробки рішення, а також її якість;
- 2) швидкість роботи запропонованих алгоритмів – проблема полягає в тому, що на великих числових послідовностях (представлення сигналів), великій кількості опорних сигналів збільшується час виконання програми, тому програмний застосунок повинен бути розроблений в парадигмі багатопотокового виконання;

3) вибір оптимального сигнального процесора із запасом обчислювальної потужності;

Для організації найбільш ефективного процесу розробки було застосовано ітеративний процес розробки, який включає наступні етапи: PoC (Proof Of Concept) застосунку (проектування та швидка реалізація), друге проектування застосунку з урахуванням помилок попереднього етапу, розробка власне застосунку.

Практична цінність кореляційного приймача із синхронізацією полягає у розробці нового та ефективного алгоритму синхронізації сигналів у приймачах. Даний варіант синхронізації в залежності від параметрів дасть можливість ефективніше використовувати канали передачі виключно для передачі корисних сигналів, а також зменшувати розміри кадрів, не навантажуючи їх системними прапорцями. Крім того, кореляційний приймач дасть змогу ефективніше боротися із завадами та виділяти корисний сигнал з них.

Наукова цінність полягає у реалізації декількох алгоритмів кореляції, щоб можна було бачити якість кожного з них. Це дасть можливість використовувати той чи інший алгоритм в залежності від умов, в яких система могла б використовуватись.

1 ОГЛЯД РІШЕНЬ ТА ТЕХНОЛОГІЙ

В загальному практично усі цифрові приймачі реалізуються на основі так званих Digital Signal Processors або цифрових сигнальних процесорах. Їхня перевага полягає в спеціалізації на обробку сигналів, математичних розрахунків тощо.

Цифрові сигнальні процесори (DSP) – спеціалізовані мікропроцесори, архітектура яких оптимізована під роботу для цифрової обробки сигналів. Основна мета у використанні таких процесорів полягає у вимірюванні, фільтрації, стисненні безперервних аналогових сигналів в режимі реального часу. Крім того, в загальному цифрові сигнальні процесори використовують для виконання алгоритмів з обробки сигналів, але не в режимі реального часу – цифрова фільтрація, перетворення Фур'є, пошук сигналів тощо. Також, сигнальні процесори мають велику перевагу – це низьке енергоспоживання, що особливо ефективно у випадку використання їх в портативних пристроях. Сигнальні процесори зазвичай використовують спеціальну модель пам'яті, яка дозволяє одночасно діставати дані або інструкції. На рисунку 3.24 зображена типова схема системи, що працює із сигналами (гітарні перетворювачі, аудіосистеми тощо).



Рисунок 1 – Типові компоненти системи для цифрової обробки [1]

Даний вид процесів використовується в наступних областях:

- кодування сигналів (аудіо та відео);
- розпізнавання голосу та зображень;
- спектральні аналізатори;
- голосові та музичні синтезатори;
- системи гідро- та радіолокації.

З математичної точки зору для сигнального процесора ставляться задачі, де потрібно необхідно здійснювати поелементне множення елементів багатокomпонентних векторів дійсних чисел, подальшого знаходження суми похідних (наприклад, обчислення при пошуку максимумів кореляційних та автокореляційних функцій вибірок сигналів). Саме тому, DSP мають оптимізовану швидкодію для виконання саме таких операцій. Фактично, сигнальні процесори орієнтовані на багатократне виконання операції множення в режимі реального часу з елементами складних масивів тощо. На рисунку 3.24 зображено базову архітектуру сигнального процесора.

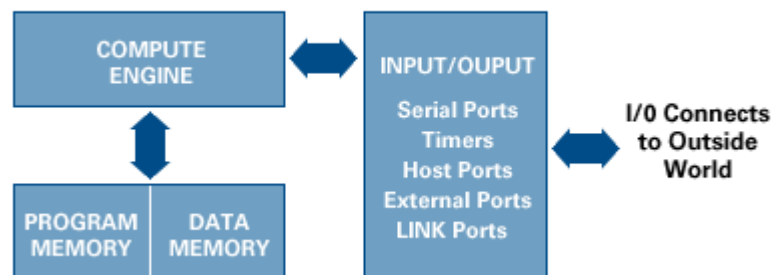


Рисунок 2 – Архітектура DSP [2]

Дана архітектура включає в себе наступні елементи:

- пам'ять програми – зберігаються програми, які сигнальні процесори використовують для обробки даних;
- пам'ять даних – містить інформацію, яку необхідно оброблювати;
- обчислювальне ядро – безпосередньо виконує математичну обробку, звертаючи до програми, що міститься в пам'яті програми, і даним, що знаходяться в пам'яті даних;
- підсистема вводу виводу – компонента, яка відповідає за взаємодію із зовнішнім світом.

Таким чином, архітектура DSP спеціально оптимізована під певні математичні операції. Першочергово сигнальні процесори орієнтовані на

багатократне множення з дуже швидким розрахунком адрес елементів масивів, що перемножуються:

- “множення з накопиченням” - реалізовано апаратно і виконується за один машинний такт;
- апаратна реалізація багатократного повторення певного набору команд;
- використання декількох портів звернення до пам’яті сигнального процесора для одночасного вибору команд та двох аргументів операцій для швидкого виконання множення з накопиченням;
- здатність до векторно-конвеєрної обробки з допомогою генераторів послідовності адрес.

1.1 Порівняння з реалізацією приймача на інших обчислювальних платформах

Реалізація приймачів може бути здійснена з використанням різноманітних пристроїв, що здатні проводити математичні обчислення в режимі реального часу. В залежності від знань, технологічних можливостей, приймачі можна будувати з використанням графічних процесорів (GPU), логічних інтегральних схем (ПЛІС) та так званих ASIC–пристроїв. Огляд таких рішень є важливим етапом в розробці приймача з на основі DSP, оскільки в залежності від поставлених задач та технічних можливостей можна зібрати різноманітні схеми для приймання будь-яких сигналів.

1.1.1 Приймачі на основі GPU

Обробка сигналів та операції над сигналами вимагають великої кількості обчислювальних потужностей. В області DSP десятиліттями використовувались спеціалізовані цифрові пристрої для обробки сигналів, які в свою чергу заточені на виконанні однієї чи двох операцій, але за дуже малий час. На жаль, навіть при такій конфігурації, у випадку великої кількості даних багатовимірних сигналів, не вистачає потужності для обробки в режимі реального часу.

В сучасний період великої популярності набирає варіант проведення обчислень на потужностях GPU. Це можливо завдяки специфічній структурі графічного процесора, який складається в основному з простих елементів, які реалізують підхід SIMD (Single Instruction Multiple Data). На рисунку Такі GPU мають можливість здійснювати операцію на великій кількості даних одночасно.

Сучасні графічні процесори можуть здійснювати тисячі одночасних обчислень в окремих потоках та процесах. Завдяки цьому, багато задач цифрової обробки сигналів можуть бути вирішені з використанням підходу “розділяй та володарюй”.

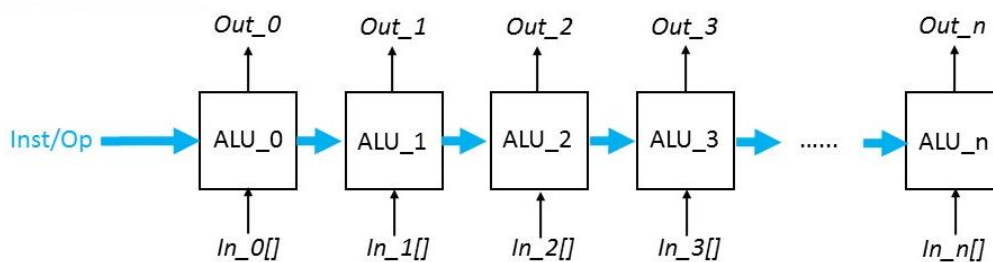


Рисунок 3 – Модель обчислення в GPU (SIMD)

В системі приймачів використання великогабаритних графічних процесорів є збитковим в рамках цілої індустрії зв'язку. Натомість бувають випадку, коли використання GPU виправдане з точки зору технічних можливостей та відсутність вимог до габаритних розмірів. Спеціалізоване програмне забезпечення CUDA дозволяє проводити різноманітні розрахунки паралельно та в режимі реального часу, що в свою чергу є однозначною перевагою GPU для обробки сигналів. Основною перевагою використання GPU перед DSP є надзвичайно більша швидкість обробки математичних операцій та більш зрозуміліший інтерфейс для програмування необхідного функціоналу. Але великі габаритні розміри у порівняння з DSP є суттєвим недоліком графічного процесора.

1.1.2 Приймачі на основі ПЛІС FPGA

FPGA це один із видів ПЛІС, суть якого полягає в тому, що логіка роботи такого електронного компоненту не визначається на етапі виготовлення, а задається засобами програмування. Цінність данного виду електронних компонент в цифровій обробці сигналів полягає у високому рівні паралелізму, який може бути реалізований засобами FPGA та програмування.

Натомість, підтримуючи високий рівень паралелізму, FPGA не здійснює обчислення над числами з плаваючою крапкою, тому у випадках де основною метою є підтримання точності, таку систему не слід використовувати. Наприклад, основною відмінністю від DSP є можливість виконання одночасного виконання необмеженої кількості операцій, забезпечуючи при цьому високий рівень паралелізму. На рисунку 3.24 зображено типову архітектуру ПЛІС типу FPGA.

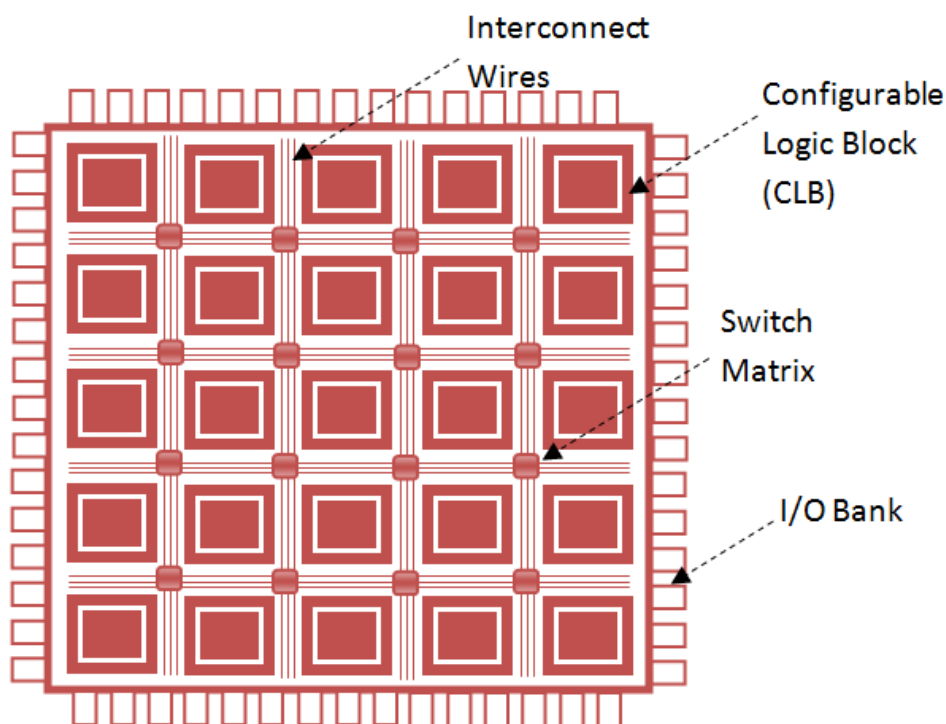


Рисунок 4 – Типова архітектура ПЛІС FPGA **Ошибка!**

Источник ссылки не найден.

Таким чином архітектура ПЛІС складається з наступних елементів:

- матриці перемикання;
- з'єднувальні лінії;
- блоки вводу/виводу;
- логічні програмовані блоки.

В свою чергу реалізацію алгоритмів можна здійснювати з використанням спеціалізованої мови програмування VHDL або мова програмування високошвидкісних інтегральних мікросхем. Всі команди, реалізовані даною мовою програмування виконуються паралельно завдяки механізму конвеєризації.

Основною особливістю такого роду обчислювальних пристроїв є конвеєрне виконання команд. Конвеєр представляє із себе обчислювальний потік, який на кожному етапі виконує якусь операцію. Крім того, в кожен момент часу виконанні, в конвеєрі знаходиться декілька команд на різній стадії виконання. Це в свою чергу дає можливість збільшити швидкість виконання математичних команд.

При паралельній обробці операцій на різних конвеєрах максимальна швидкість досягається у випадку виконання однотипних команд, які не залежать одна від одної. У випадку, якщо команди залежать одна від одної, то в конвеєрі використовуються такти очікування.

Фактично в рамках приймача, даний пристрій буде відігравати роль процесора розрахунку математичних операцій.

Головною перевагою FPGA перед DSP це є однозначно більший рівень паралелізму та відповідно більша швидкість обробки математичних операцій та даних із сигналу. За розмірами FPGA приблизно такі самі як DSP, тому це ідеальне рішення для систем де необхідний високий паралелізм роботи. Це відповідно і найбільша перевага даної системи. При цьому, складність у програмування деяких алгоритмів математичної обробки сигналів для ПЛІС можуть через специфіку мови програмування можуть стати перешкодою для розробки приймачів.

1.1.3 Приймачі на основі ASIC

Найбільшу популярність дана технологія отримала після криптовалютного буму як пристрій для здійснення майнінгу і заробітки криптовалют. Головною перевагою ASIC є вузька спеціалізація на здійсненні невеликої кількості задач. Завдяки цьому, такі пристрої можуть використовуватись для обробки цифрових сигналів. Існують ASIC пристроїв для обчислення різних математичних операцій – перетворення Фур'є, знаходження розгортки, кореляції тощо. На рисунку 3.24 зображено зовнішній вигляд ASIC плати. Кожен радіатор покриває окремий чіп, що спеціалізується на певній операції. В даному випадку, на рисунку представлена фотографія пристрою для майнінгу. У випадку приймача кількість цих чіпів зменшується в 10 разів, оскільки немає потреби у великій кількості

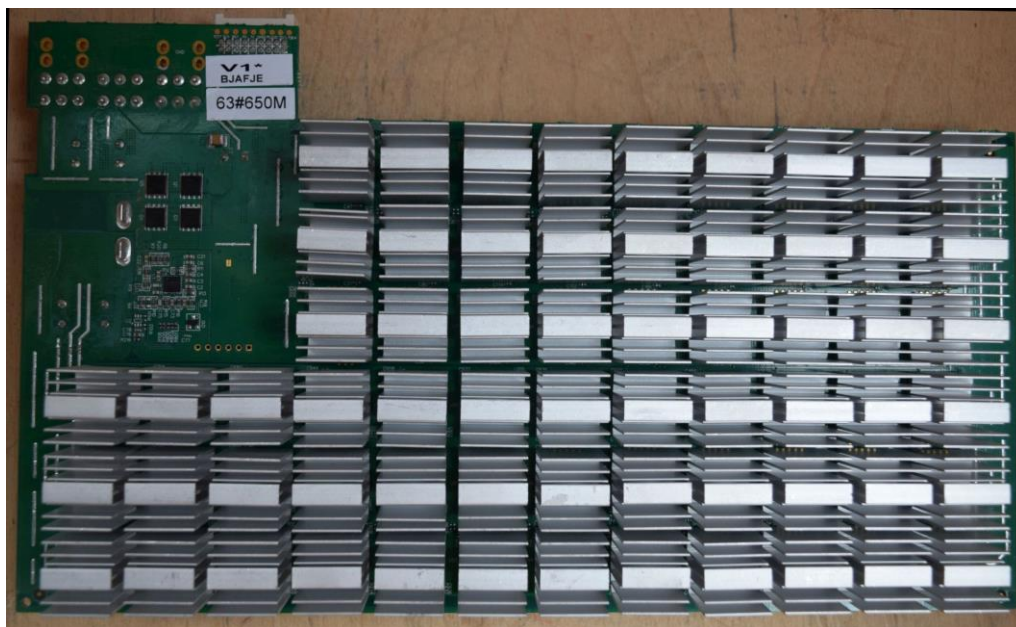


Рисунок 5 – Зовнішній вигляд плати ASIC

обчислень.

1.2 Програмне забезпечення для реалізації моделей

1.2.1 Бібліотека Numpy

Дана бібліотека є основною та важливою бібліотекою, якою користуються в сферах DSP, Data Science, математики, фізики тощо. Бібліотеку Numpy можна розглядати як аналог застосунку Matlab. Дана бібліотека вміщає в собі величезну кількість математичних операцій як над скалярами, так і над векторами та матрицями. Конкретно для області DSP Numpy корисна як математична основа для кореляції, модуляції та генерація різних форм сигналів.

1.2.2 Бібліотека Scipy

Окремий модуль, який починався як частина бібліотеки Numpy, але історично виокремився в окрему бібліотеку. Дана бібліотека широко застосовується в наступних сферах:

- обробка сигналів;
- обробка зображень;
- робота з генетичними алгоритмами;
- обчислення інтегралів функцій та багато іншого.

1.2.3 Багатопотокове виконання

Багатопотоковість це фундаментальний підхід для ефективної реалізації програмних застосунків, що працюватимуть на багатоядерній архітектурі сервера чи ПК. Таким чином, існують дві найпоширеніші причини використання багатопотоковості – по-перше, збільшити ефективність використання процесорного часу на багатоядерній архітектурі сучасних CPU, по-друге, у випадку, коли певна логіка програми повинна виконуватись паралельно або частково асинхронно. Надалі, контекст розуміння багатопотоковості буде обмежено мовою програмування Python, як основного інструменту розробки програмного комплексу в рамках магістерської дисертації.

Основі обмеження, що накладаються мовою Python є Global Interpreter Lock (GIL). GIL це механізм, який дозволяє запускати в один момент часу процесором може виконуватись тільки один потік. Це в свою чергу було зроблено для недопущення конфлікту за ресурси, а саме боротьби за окремі змінні. Один потік, що виконується має повний доступ до всього оточення в рамках програми, що в свою чергу підвищує потокозахищеність та спрощує роботу з багатопотоковим виконанням.

Таким чином, можна навести порівняльну таблицю 2 з плюсами та мінусами використання багатопотоковості в мові Python.

Таблиця 1 – Переваги та недоліки багатопотокового виконання на мові Python

Переваги	Недоліки
Збільшення швидкодії за рахунок використання ядер процесора	На однопроцесорній системі багатопотокове виконання може навпаки погіршити швидкодію системи
Завдяки багатопотоковості можна створювати user friendly графічний інтерфейс, оскільки основі обчислення будуть здійснюватись у фоні програми	Необхідна синхронізація, щоб уникнути взаємного виключення при доступі до спільних ресурсів.
Глобальні змінні – якщо в рамках програми зміниться значення глобальної змінної, то потоки однозначно знатимуть про це, тому в цьому випадку потрібно менше думати про синхронізацію даних між	Складно налагоджувати багатопотокову програму, підвищується складність застосунку в цілому.

потоками. Також потоки можуть мати власні локальні дані.	
--	--

Багатопроцесорні додатки мовою Python можна реалізовувати з використанням різноманітних модулів. Наприклад, модуль `subprocess`. В даному випадку все, що виконується даним модулем, виконується в окремому процесі, а не потоці. Але в даному підході є багато проблем. Наприклад, головна проблема полягає в тому, що надзвичайно складно передавати дані між процесами. Для реалізації подібного механізму дані довелося би зберігати окремо в пам'яті в серіалізованому вигляді, а це в свою чергу ускладнює програмний застосунок.

Крім того, існує модуль `multiprocessing`, який більше схожий на модуль `threading`, але при цьому є декілька відмінностей. Замість потоків також створюються окремі процеси, але для синхронізації даних між процесами використовуються зовсім інші механізми. Наприклад, черги (`queue`) та канали (`pipe`). При цьому, також існують аналоги `lock`'ів, семафорів, які присутні в `threading`. Також, даний модуль надає механізм роботи зі спільною пам'яттю. Також, бібліотека `multiprocessing` дає можливість створювати пул процесів, замість мануального створення окремих процесів в кодовій базі.

1.3 Огляд існуючих рішень для моделювання приймачів

1.3.1 GNU Octave (Signal Processing)

Даний програмний комплекс є відкритим аналогом застосунку Matlab. Підтримує скрипти створені в середовищі Matlab, що в свою чергу дозволяє без проблем запускати Matlab скрипти на будь-якому середовищі операційної системи. Все таки головною перевагою даного програмного комплексу є кросплатформність та підтримка Matlab. На рисунку 1.4 зображено графічний інтерфейс користувача GNU Octave із введенням скриптом для відображення взаємнокореляційної функції, що в свою чергу буде відображено на рисунку 1.5.

Даний застосунок є потужним комплексним аналогом Matlab для математичних обчислень, але через функціональні причини, кращим варіантом

буде вивчення та використання Matlab, а GNU Octave використовувати як середовище виконання для платформ де Matlab не підтримується.

Крім того, наявність CLI в GNU Octave дає можливість створювати автоматизовані процеси в якості демонів чи процесів, які необхідно виконувати через термінал. Натомість, Matlab такої можливості не має.

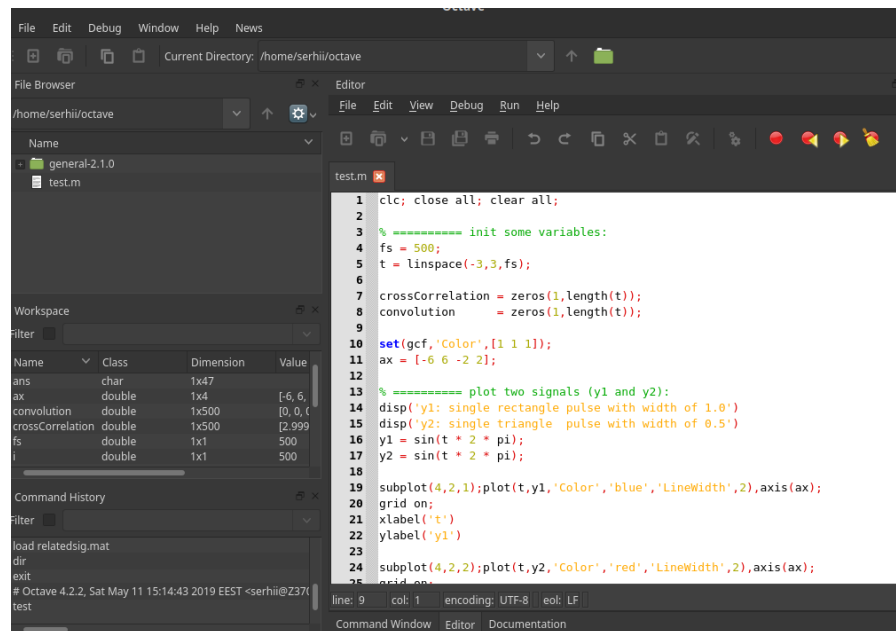


Рисунок 6 – Графічний інтерфейс користувача GNU Octave

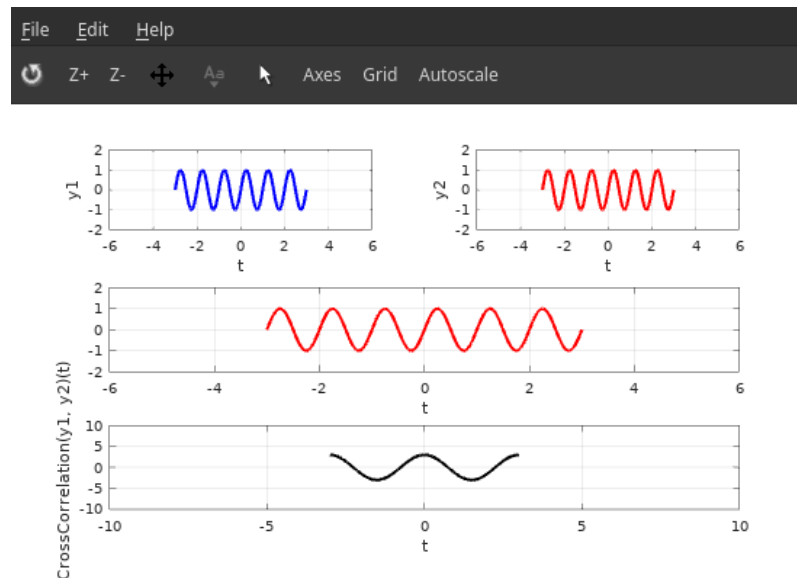


Рисунок 7 – Візуалізація процесу знаходження кореляції між двома синусоїдами

1.3.2 Scilab

Альтернатива Matlab, яка поширена як відкрите програмне забезпечення. Містить в собі альтернативу Simulink – Scicos, що в свою чергу збільшує кількість можливостей, порівняно з GNU Octave. Суттєвим недоліком даного застосунку є відсутність підтримки виконання скриптів із Matlab. Можливість розширення базового функціоналу створенням власних модулів мовою C/C++. Для підтримки виконання коду Matlab необхідно встановлювати утиліти для конвертації коду Matlab в Scilab. Також, наявний CLI інтерфес для виконання програм з терміналу, що також дає широкий спектр для автоматизації математичних обчислень. На рисунку 3.24 зображено графічний інтерфейс користувача застосунку Scilab.

Перевагами даного програмного комплексу є наявність утиліти для візуального моделювання, аналог Simulink, Scicos. Але ця перевага не покриває повністю застосунок Simulink комплексу Matlab через відсутність повної документації та меншої кількості пакетів з функціональними блоками для роботи

і моделювання в різних сферах, де потребуються математичні обрахунки. В області цифрової обробки сигналів, даний пакет містить дуже мало блоків для моделювання, тому даний програмний застосунок не підходить для задач

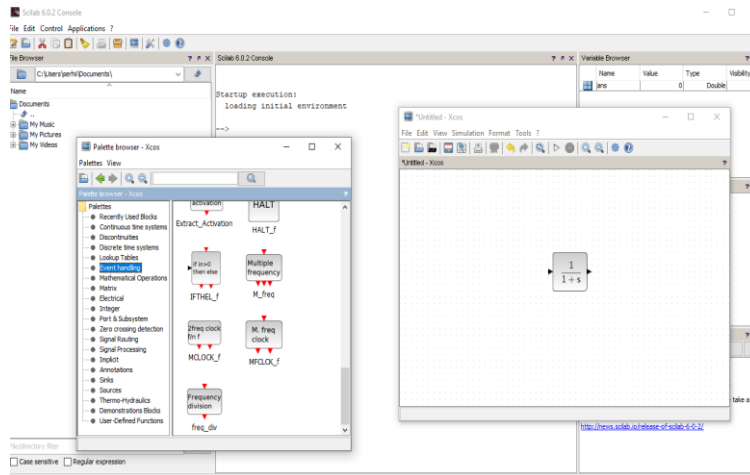


Рисунок 8 – Графічний інтерфейс користувача застосунку Scilab

створення моделей тощо.

Незначною перевагою Scilab є малий розмір інсталяційного пакету, завдяки чому час встановлення та зайнятого місця стає дуже малим у порівнянні з Matlab.

1.3.3 Програмний застосунок для математичних моделювання Matlab/Simulink

Найпоширеніший застосунок для математичних моделювань, цифрової обробки сигналів тощо. З-поміж решти застосунків виокремлюється багатим графічним інтерфейсом та великою кількістю інструментів моделювання, великою кількістю прикладів виконання тих чи інших моделювань, їхнього теоритичного опису, а також дуже багата довідка під кожен елемент інструментів емуляції та моделювання. На рисунку 3.24 зображено графічний інтерфейс користувача, де зображено зліва поточна директорія у файловій системі, в центрі – робоче поле з редактором скриптів та термінал для вводу команд та прикладів.

Справа зображено поточний стан даної сесії в програмі, де відображаються значення змінних, пам'яті тощо.

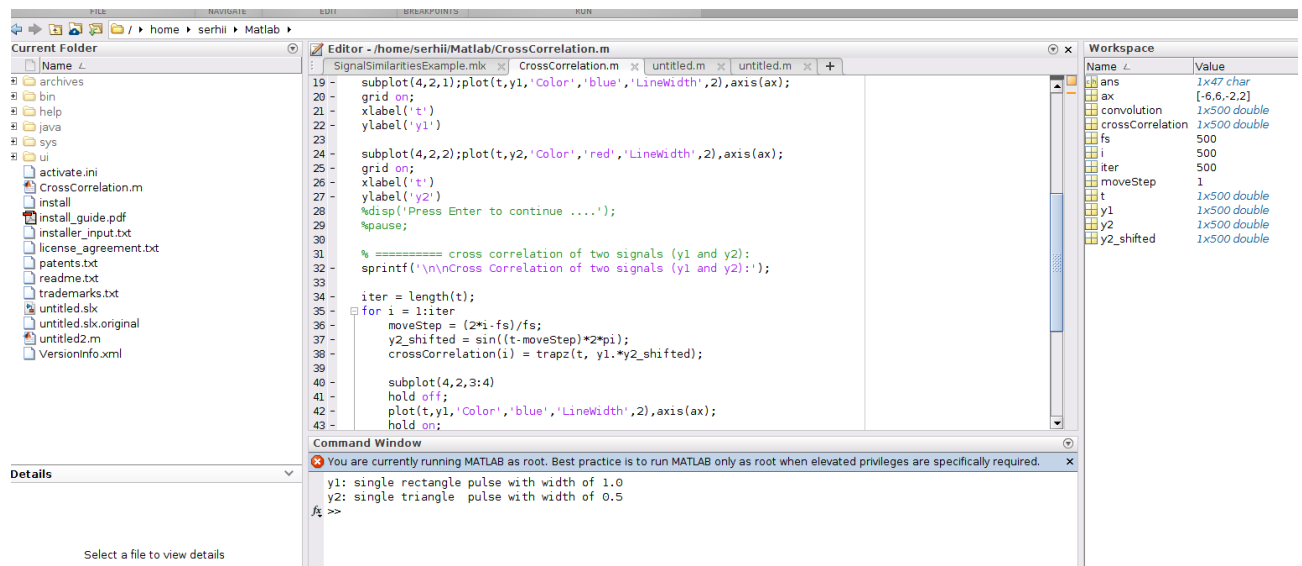


Рисунок 9 – Графічний інтерфейс робочого поля Matlab

На рисунку 3.24 зображено графік, що візуалізує в режимі реального часу процес знаходження АКФ для двох синусоїд. Відображення графіків практично нічим не відрізняється від інших інструментів моделювання, що були розглянуті вище.

Однією із основних переваг Matlab є наявність пакету для візуального моделювання різноманітних процесів. Крім того, наявність пакету DSP також відкриває доступ до великої кількості прикладів Simulink з цифрової обробки сигналів, що свою чергу якісно піднімає рівень моделювання обробки сигналів та їх синхронізації.

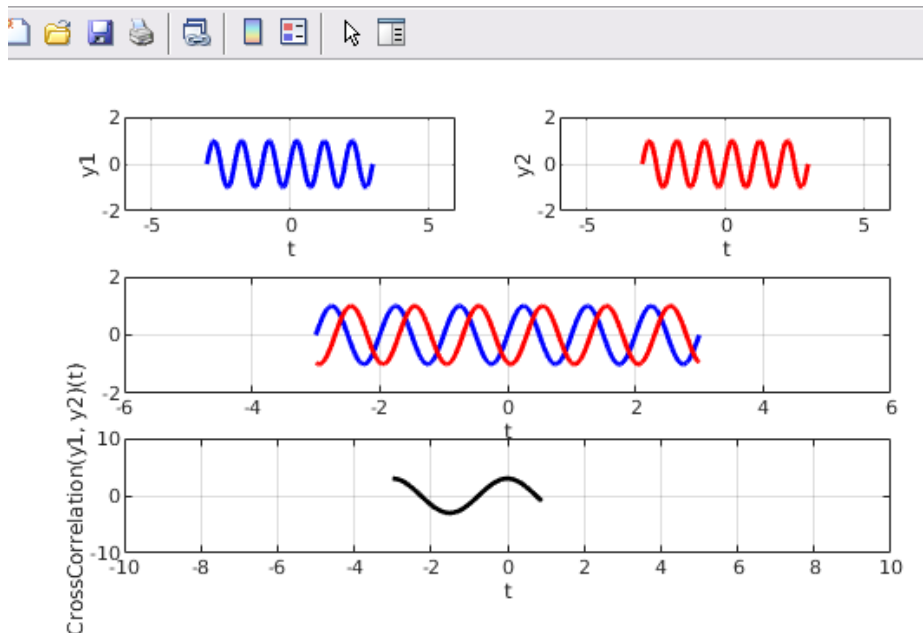


Рисунок 10 – Візуалізація процесу знаходження кореляція між двома синусоїдами

До недоліків програмного комплексу Matlab слід віднести складність програмного застосунку у порівнянні з іншими, займає багато дискового простору та на UNIX платформах здійснює високе навантаження на обчислювальні ресурси ПК.

2 ТЕОРЕТИЧНИЙ ОПИС РІШЕННЯ

2.1 Кореляція

Під кореляцією розуміють взаємозв'язок двох або декількох випадкових величин. Основна суть полягає в тому, що при зміні значення однієї змінної закономірно змінюється іншої змінної. При розрахунку кореляції намагаються визначити, чи існує статистичний зв'язок між двома або декількома змінними в певній вибірці даних.

Таким чином, значний рівень кореляція між двома випадковими величинами завжди вказує на певний статистичний зв'язок в даній вибірці, але при цьому зв'язок не завжди повинен існувати в іншій вибірці даних, а тим більше мати причинно-наслідковий характер. Крім того, відсутність кореляції між величинами не завжди вказує на те, що між величинами немає зв'язку. Тоді в цьому випадку говорять про те, що величини мають зв'язок нелінійного характеру.

Для визначення кореляції обраховують кореляційний коефіцієнт, який є чисельним представленням статистичної залежності. Тому, зазвичай, за значенням коефіцієнту і визначають рівень зв'язку. Наприклад, деякі кореляційні коефіцієнти можуть набувати значень із проміжку $-1..1$. Якщо коефіцієнт кореляції в даному випадку становить 0, то це означає, що ніякого зв'язку між величинами немає. Якщо ж значення коефіцієнту тяжіє до -1 або 1 , то це явно вказує про наявність певного зв'язку між величинами, при цьому відрізняється лише характер залежності:

- додатне значення кореляційного коефіцієнту показує, що збільшення або зменшення однієї величини закономірно збільшує або зменшує іншу величину. Фактично, пряма лінійна залежність типу “збільшення-збільшення” або “зменшення-зменшення”;
- від'ємне значення кореляційного коефіцієнту показує, що збільшення або зменшення однієї змінної зменшує або збільшує відповідно другу змінну.

Для знаходження кореляційних коефіцієнтів існують так звані методи кореляційного аналізу, які включають:

- інтервальна кореляція;
- рангова кореляція – метод кореляційного аналізу, що полягає в знаходженні відношень між змінними, які відсортовані за зростанням значення. Найчастіше даний метод викиористовуєть у випадках пошуку зв'язку між якісними ознаками;
- дихотомічна кореляція (кореляція дихотомічних змінних);

2.1.1 Коефіцієнт кореляції Пірсона

Даний кореляційний коефіцієнт використовується для визначення зв'язку між двома змінними, які вираховані в метричних шкалах на одній і ті самій вибірці. Цей коефіцієнт дозволяє визначити рівень пропорційності мінливості двох змінних. Величина даного коефіцієнту кореляції приймає значення від 0 до 1. Перед початком розрахунку даного коефіцієнту необхідно спочатку виконати наступні умови:

- досліджувані змінні повинні бути нормально розподілені;
- досліджувані змінні повинні бути виміряні в інтервальній шкалі;
- кількість значень в змінних повинна бути однаковою.

Даний коефіцієнт має декілька слабких сторін:

- нестійкий до викидів;
- можна визначити тільки силу лінійного взаємозв'язку між змінними, інші види взаємозв'язку виявляються з допомогою методів регресійного аналізу.

На рисунку 3.24 зображено приклад діаграми розсіву з різними значенням коефіцієнту кореляції.

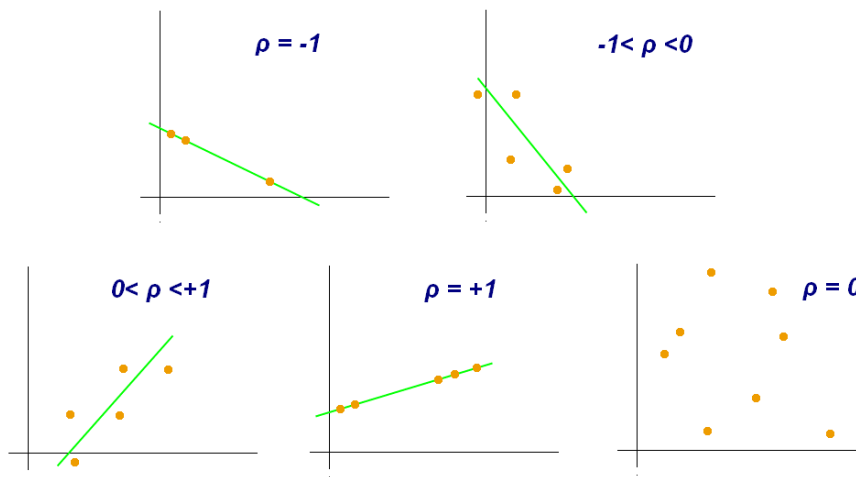


Рисунок 11 – Діаграма розсіву для кореляційного коефіцієнту Пірсона

2.1.2 Коефіцієнт рангової кореляції Спірмена

Коефіцієнт рангової кореляції Спірмена використовується для знаходження та оцінки близькості зв'язку між двома рядами кількісних показників. Якщо в обох рядах, ранги показників, що відсортовані за зростанням чи спаданням, в більшості випадків співпадають, то робиться висновок, що існує прямий кореляційний зв'язок. Якщо ж ранги мають протилежний напрямок зв'язку, то це означає, що між показниками існує обернений кореляційний зв'язок.

За своїми властивостями коефіцієнт Спірмена приблизно схожий на коефіцієнт Пірсона, а саме:

- коефіцієнт приймає значення від -1 до 1, де -1 означає про обернений зв'язок, а 1 – прямий зв'язок;
- якщо кореляційний коефіцієнт дорівнює 0, то зв'язок між величинами повністю відсутній;
- чим ближче модуль коефіцієнта кореляції до одиниці, тим сильнішим є зв'язок між величинами, які вимірюються.

При використанні даного кореляційного коефіцієнту оцінка зв'язку відбувається наступним чином: умовно існують деякі проміжки значень по модулю за якими можна однозначно про тісноту зв'язків; наприклад, значення,

що лежать в проміжку між 0 та 0.3 – індикатор слабкого зв'язку, значення між 0.4 та 0.7 – показники середнього рівня зв'язку, а значення між 0.7 та 1 – однозначно сильний зв'язок між змінними.

2.2 Синхронізація приймача

У всіх схемах, які займаються кодуванням бітів у символи, приймач повинен знати, коли набір символів закінчується та починається передача наступного набору. У випадках коли оцінюється швидкодія приймачів будь-якого виду завжди припускається існування певного виду синхронізації сигналів, що приймаються. Наприклад, у випадку PSK припускається, що приймач може генерувати набір опорних сигналів з однією фазою, яка у свою чергу ідентична фазі елементів сигнального алфавіту передавача. В процесі синхронізації, приймач здійснює порівняння прийнятого символу (або послідовності символів) з опорними сигналами і обирається найбільш схожий.

Крім варіанту порівняння з опорним сигналом існують варіанти з використанням самосинхронізуючих кодів (фізичне кодування), наприклад – манчестерське кодування, код з поверненням до нуля (RZ код) тощо.

Також, одна із стратегій синхронізації полягає в тому, щоб надіслати до приймача окремий сигнал годинника. Окрема шина для годинника – це достатньо затратний варіант синхронізації у випадку великих мереж, і таку реалізацію доцільно використовувати у випадку комп'ютерних шин, або в мережі з короткими кабелями. Тому для вирішення цієї проблеми створений наступний варіант: достатньо змішати сигнал годинника із сигналом даних та об'єднати їх з допомогою операції XOR. Даний підхід реалізовано в Манчестерському кодуванні.

Кожен з підходів реалізації синхронізації сигналів використовується в залежності від типу передачі даних, а саме: асинхронна та синхронна передачі даних.

2.2.1 Асинхронна та синхронна передачі даних

При асинхронній передачі дані передаються в канал передачі даних у вигляді послідовності бітів, з якої при прийомі треба виділити інформаційні байти для подальшої обробки. В такому випадку, кожен байт представляє набір системних прапорців (start та stop біти) та інформаційної частини. Таке розділення дозволяє провести розділення між кожним переданим байтом в каналі передачі. Такий підхід дозволяє сповіщати сторону приймача про початок передачі даних. На рисунку 3.24 зображено часову діаграму для асинхронної передачі даних: можна побачити start біт (синім кольором) та stop біт (червоним кольором)

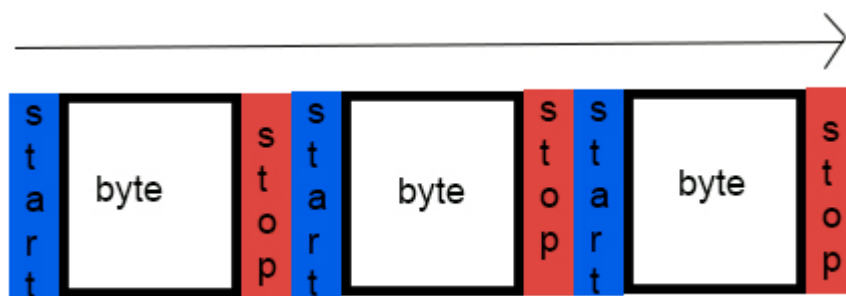


Рисунок 12 – Часова діаграма асинхронної передачі даних

Крім того, асинхронна передача обумовлена наступними факторами:

- погана якість каналу передачі даних (фізичне середовище для передачі) – багато завад тощо;
- здешевіння апаратури;
- підвищення надійності передачі даних.

Все забезпечується використанням синхронізації на рівні байтів. Таким, чином в асинхронних системах передачі даних відбувається лише побітова синхронізація, упускаючи кадрову синхронізацію.

При синхронній передачі даних набори бітів формуються у вигляді кадрів, які обрамляються байтами синхронізації. Стартові чи стоп біти перед та після байтів відсутні, натомість включаються в передачу байти синхронізації, який

представляє із себе заздалегідь відому комбінацію бітів. Цей байт потрібен для сповіщення приймача про початок передачі. На рисунку 3.24 зображено часову діаграму синхронної передачі даних, де зеленим кольором зображено байти синхронізації (початок та кінець передачі).

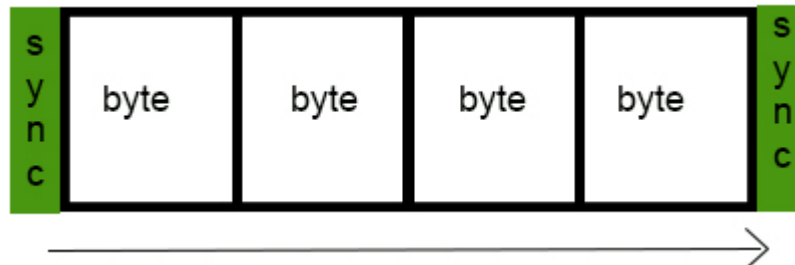


Рисунок 13 – Часова діаграма синхронної передачі даних

2.2.2 Синхронізація з використанням опорних сигналів

При генерації опорних сигналів приймач повинен синхронізуватись з несучою хвилею, що приймається. Це означає, що фаза несучою, що потрапляє на приймач, та фаза опорного сигналу в приймачі повинні бути узгоджені. У випадку, якщо у несучому сигналі на вході приймача не закодовано інформацію, то відповідно опорний сигнал та вхідний несучий сигнал будуть проходити одночасно через нуль. Цей процес називається фазовим автопідлаштуванням частоти.

Бувають випадки, коли сигнал модулює безпосередньо не несучий сигнал, а піднесучий, то в такому разі необхідна визначити як фазу несучого так і піднесучого. Якщо передавач не займається синхронізацією фаз несучого та піднесучого сигналів, то необхідною умовою є генерація копії піднесучого сигналу, але при цьому керування фазою піднесучого сигналу здійснюється окремо від несучого. Це в свою чергу дозволяє приймачу отримати фазову синхронізацію по обох видах несучого сигналу.

Символьна та фазова синхронізація мають спільну рису – при синхронізації в приймачі генерується копія частини переданого сигналу. У

випадку фазової синхронізації це є копія несучого сигналу, а для символної – меандрова хвиля

2.2.3 Фізичне кодування та самосинхронізуючі коди

Спосіб синхронізації при синхронній передачі даних, під час якого синхронізація дискретизованого сигналу передавача і дискретизованого сигналу приймача відбувається в самому сигналі за рахунок використання специфічного кодування.

Специфічне кодування – це самосинхронізуючі коди, які надають приймачу інформацію про те, коли саме необхідно виконати розпізнавання біта інформації (або порції бітів). В таких видах кодування будь-який різкий перепад фронту сигналу слугує ефективним маркером, який вказує на початок синхронізації приймача та передавача.

До самосинхронізуючих кодів відносять:

- RZ (return to zero) код – один із способів лінійного кодування на фізичному рівні. Формування сигналу відбувається за наступним правилом: код є 3-х рівневим, що дає можливість коду повертатись в нульове значення після передачі значущого інтервалу. Інформаційний перехід відбувається з самого початку значущого інтервалу, повернення до нульового рівня відбувається в середині значущого інтервалу, а далі рівень залишається нульовим до кінця значущого інтервалу. Найважливіше, що в цьому коді повернення до нуля забезпечує синхронізацію сигналу передавача та приймача. На рисунку 3.24 зображено часові діаграми коду;
- біполярний AMI код – вид фізичного кодування. Так само як і RZ код являється 3-х рівневим: при поданні на вхід логічної одиниці змінюється потенціал на верхній або нижній, в залежності від попереднього рівня сигналу, на якому передавалася логічна одиниця.

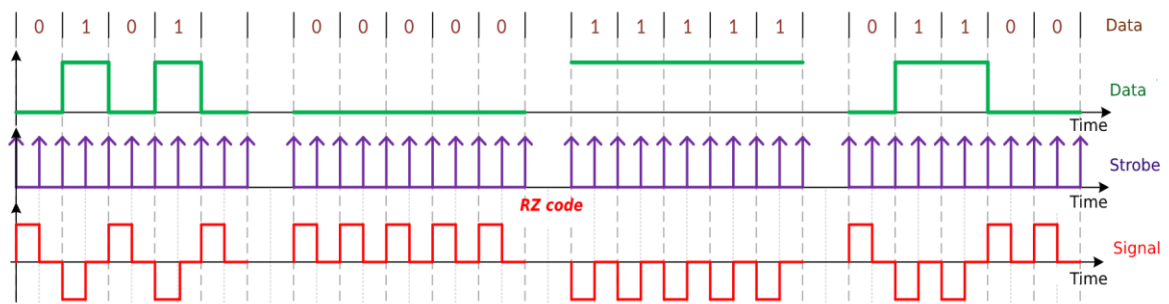


Рисунок 15 – Часова діаграма RZ коду [3]

Синхронізація відбувається за рахунок передачі зміни логічної одиниці і логічного нуля. На рисунку 3.24 відповідно зображено часову діаграму даного коду;

— NRZI (non return to zero invertive) код – розрізняють два варіанти сформованих сигналів:

а) 1-й варіант (1 option) – якщо на вхід пристрою потрапляє логічна одиниця то потенціал в каналі зв'язку змінюється, і навпаки у випадку

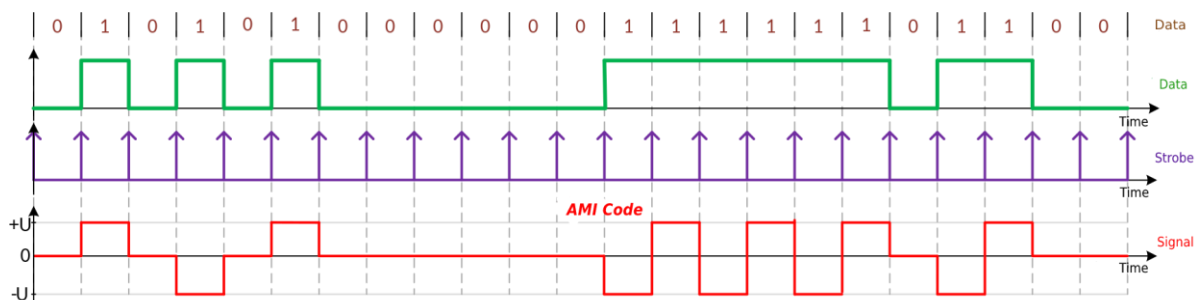


Рисунок 14 – Часова діаграма AMI коду [4]

логічного нуля на вході;

б) 2-й варіант (2 option) – такий самий варіант як 1-й тільки навпаки.

Синхронізація приймача та передавач відбувається наступним чином: фіксується передача або кінець передачі, коли на вхід потрапляє логічна одиниця, а для ефективнішого використання в закодованому повідомленні слід використовувати максимально великі послідовності

логічних одиниць (нулів) та не використовувати довгі послідовності логічних нулів (одиниць);

— Манчестерський код (за стандартом IEEE 802.3) – кодування при якому логічний нуль кодується перепадом напруги з верхнього рівня на нижній, а логічна одиниця кодується перепадом напруги з низького рівня на верхній рівень. На рисунку 3.24 зображено часові діаграму, яка

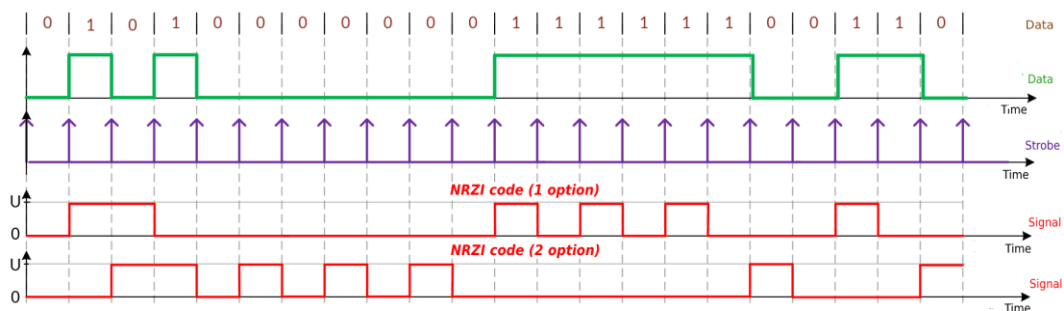


Рисунок 16 – Часова діаграма NRZI коду [5]

наочно показує роботу коду;

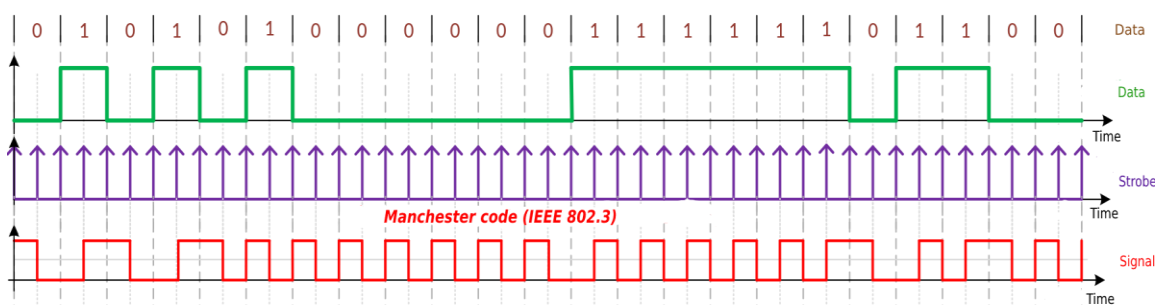


Рисунок 17 – Часова діаграма Манчестерського коду [6]

— потенційний код 2B1Q – спосіб кодування полягає у встановленні рівня сигналу у відповідність з двома бітами, що потрапляють на вхід приймача. Кожній двійці бітів відповідає потенційний рівень сигналу і всього таких рівнів 4. Синхронізація відбувається тільки в тому випадку,

якщо відсутні послідовності однакових пар бітів. На рисунку 3.24

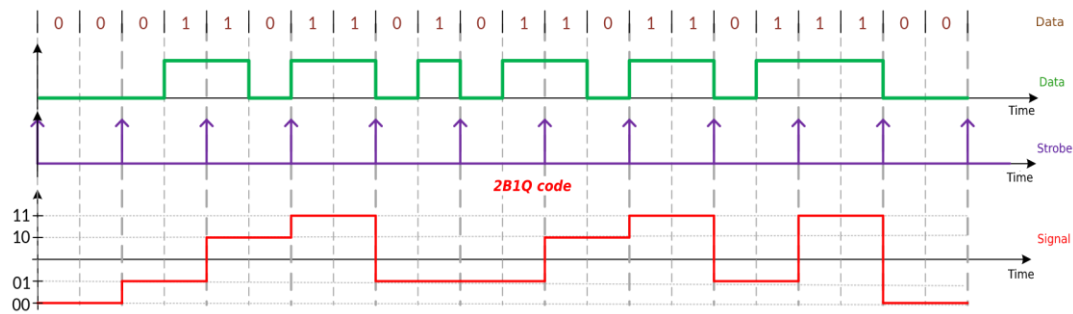


Рисунок 18 – Часова діаграма 2B1Q коду [7]

зображено часову діаграму даного кодування.

Розглянуті в даному розділі види фізичного кодування володіють властивостями самосинхронізації, що має свої переваги та недоліки. Часто в мережах з великою швидкістю передачі даних (понад 100 Мбіт/с) дані види кодування не використовуються через специфічні вимоги до каналу передачі даних.

2.2.4 Символьна синхронізація

Будь-які цифрові приймачі повинні вміти синхронізуватись із сигналами, які приходять із каналу зв'язку [15]. Ці входні сигнали є двійковими низькочастотними сигналами. Крім того, приймачу не відомо про послідовність елементів сигналу, що надходять до нього. Це так звана синхронізація без апріорних знань про інформаційний потік, що приймається.

В свою чергу пристрої зі використанням символною синхронізації поділяються на дві групи, а саме:

- розімкнуті синхронізатори;
- замкнуті синхронізатори.

2.2.4.1 Розімкнуті синхронізатори

Даний клас синхронізаторів працює наступним чином: виділяється копія виходу генератора тактових імпульсів передавача безпосередньо із інформаційного сигналу. Ці синхронізатори генерують компоненту частоти зі швидкістю передачі кадрів, при цьому пропускають вхідний низькочастотний сигнал через певну послідовність узгоджуваного фільтра та нелінійного пристрою. Необхідна компонента частоти виокремлюється з допомогою смугово фільтру, а далі компонента проходить через функцію *signum*, яка надає необхідну форму сигналу компоненти. Після даних кроків, повинен вийти прямокутний сигнал генератора тактових імпульсів.

2.2.4.2 Замкнуті синхронізатори

Замкнуті синхронізатори порівнюють вхідний сигнал з тактовими імпульсами генератора в приймачі і після цього синхронізується локальний сигнал із вхідним сигналом. Найбільш популярним видом замкнутого синхронізатора є синхронізатор з випереджаючим та запізнілим стробуванням. Його роль полягає тому, що він здійснює два окремих інтегрування енергії вхідного сигналу за двома різними проміжками символного інтервалу. Перше інтегрування починається на початку періоду передачі символу і закінчується через певний заданий час. Друге інтегрування, яке ще називається запізнілим, виконується із затримкою на певну кількість секунд і закінчується в кінці проміжку передачі символу. В свою чергу, різниця абсолютних значень виходів обох інтегруючих пристроїв є мірою визначення помилки синхронізації символів у приймачі. Це значення помилки, також, може бути використане у зворотному зв'язку для корегування прийому.

2.2.5 Кадрова синхронізація

В мережах дані передаються так званими групами або наборами бітів, іншими словами мають кадрову структуру. Як відомо комп'ютерні дані зазвичай

розбиваються на слова, які складаються з певного набору байтів, які в свою чергу комбінуються у файли, пакети, кадри тощо [15]. Таким чином, в мережах дуже важливо синхронізуватись із кадровою структурою потоку даних, щоб правильно зчитати корисну інформацію.

Найпростішим способом синхронізації передавача та приймача є передача спеціального маркера. Такий спосіб синхронізації розглянуто у розділі 2.2.1. Приймач в свою чергу повинен знати, що саме цей маркер є індикатором початку кадрової послідовності. Приймач здійснює перевірку кореляції потоку вхідних даних з опорним сигналом протягом певного відрізка часу. Якщо приймач не синхронізовано, то значення кореляції буде низьким, і навпаки, у випадку повної синхронізації кадрів кореляція буде мати найбільше значення. Підхід з використанням маркерів (start біт, stop біт тощо) має єдину перевагу – його простота. Маркером може бути лише один біт і цього буде достатньо у випадку, якщо перед прийняттям рішення про синхронізацію було зроблено достатню кількість кореляцій. Але при цьому кількість кореляцій може бути достатньо велика, що в свою чергу збільшує час на синхронізацію. Це є основним недоліком даного підходу. Такий режим синхронізації не підходить для систем, де передаються окремі пакети даних, або де потрібно здійснити швидку кадрову синхронізацію. Використання маркерів також використовуються у самосинхронізуючих кодах, що описані в розділі 2.2.3.

Натомість, в системах з нестійкою передачею даних, або де кадрова синхронізація повинна здійснитись миттєво, використовують так звані синхронізуючі кодові слова – набір бітів або байтів, які слугують індикатором початку передачі. При такому алгоритмі синхронізації, приймач постійно повинен шукати кодове слово в послідовності з використанням того ж корелятора (узгодженого фільтра). Головною перевагою даного способу є миттєва синхронізація, а недоліками є:

— затримки у знаходженні кодового слова;

— кодове слово, яке обирається для індикації початку передачі, може бути достатньо довгим, щоб виключити ймовірність помилкової детекції початку передачі.

Відповідно, підходящим кодовим словом для початку синхронізації є те слово, яке має мале абсолютне значення побічних максимумів графіка кореляції. Під побічним максимумом кореляції розуміється значення кореляції кодового слова на саме себе, але зі зміщенням по фазі. Таким чином, значення кореляції зміщеного кодового слова із самим собою вираховується за наступною формулою:

$$C_k = \sum_{j=1}^{N-k} X_j X_{j+k}. \quad (1)$$

На рисунку 3.24 зображено 9-ти бітову послідовність та значення коефіцієнту кореляції над кожним графіком. Можна побачити, що найбільший побічний максимум зображено на найпершому графіку після графіка сигналу. Значення кореляції в даному випадку становить 32, а для решти випадків 0.

Слід пам'ятати, що при використанні кодових слів існує ймовірність помилкової синхронізації. Тому, приймачі, що працюють на основі кодових слів, мають наступні характеристики – ймовірність пропущеного моменту початку кадру та ймовірність помилкової тривоги. Таким чином, при розробці подібного пристрою необхідно підтримувати низький рівень ймовірності обох характеристик.

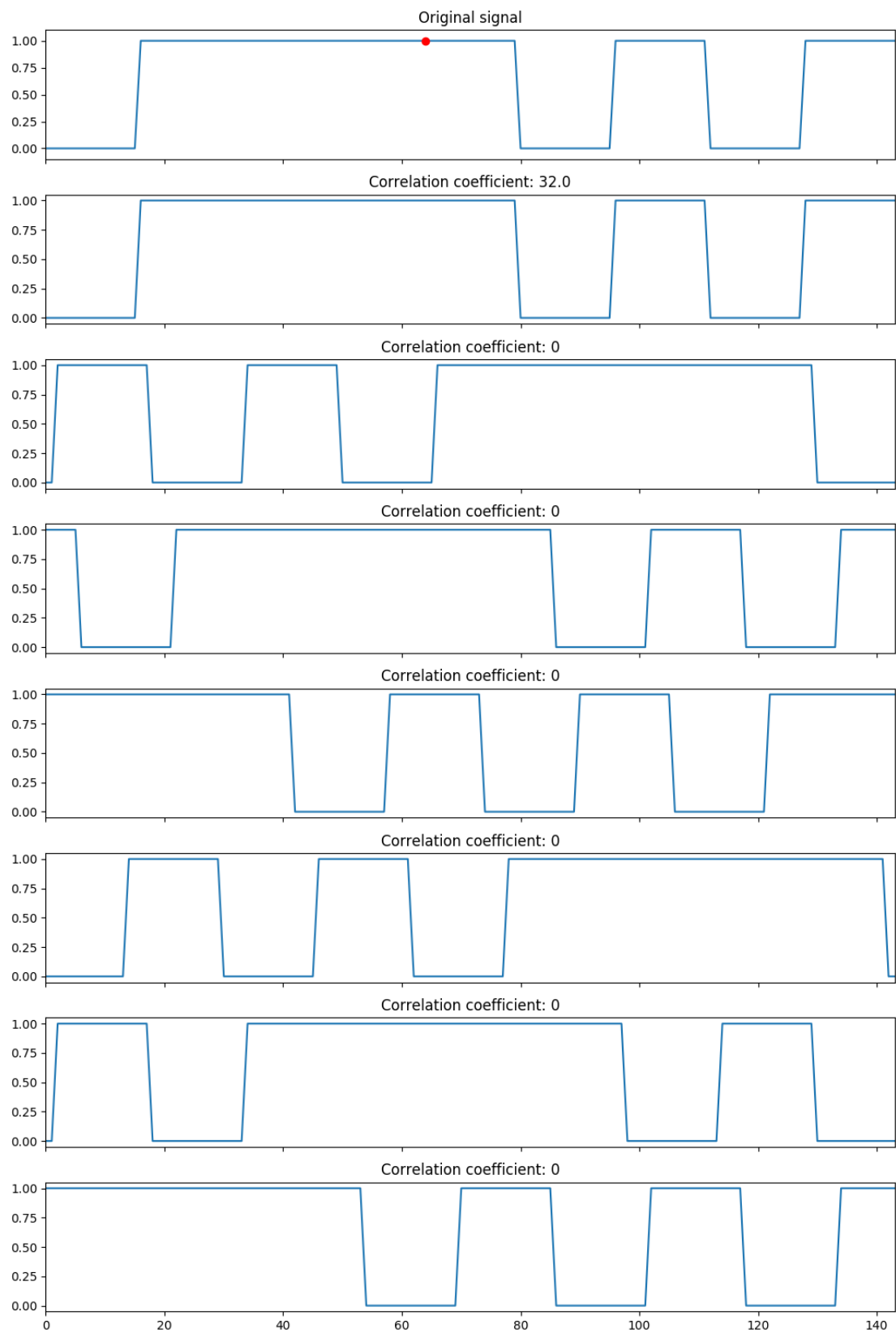


Рисунок 19 – Обрахунок кореляції кодового слова із кодовим словом
зі зміщенням

В такому випадку, щоб зменшити ймовірність пропускання моменту початку кадру, приймач повинен допускати не достатньо чітку кореляцію сигналу на вході, а це означає, що приймач повинен приймати кадр навіть у випадку наявності помилок. Крім того, такий підхід збільшує кількість символів, які потенційно можуть бути прийняті приймачем, а це в свою чергу збільшує ймовірність помилкової тривоги. Отже, дані характеристики можна описати наступними формулами ймовірностей:

$$P_m = \sum_{j=k+1}^N \binom{N}{j} p^j (1-p)^{N-j}, \quad (2)$$

$$P_{FA} = \sum_{j=0}^k \frac{\binom{N}{j}}{2^N}. \quad (3)$$

Відповідно, формула **Ошибка! Источник ссылки не найден.** вказує на ймовірність пропущення слова довжиною N біт, у випадку, коли k – максимальна кількість помилок, яку може пропустити детектор приймача, а p – ймовірність виникнення помилки в біті; формула **Ошибка! Источник ссылки не найден.** – ймовірність помилкової тривоги.

З формул ймовірності спостерігається наступна залежність: коли k зростає та p достатньо мале, то P_m зростає. При цьому зі збільшенням kP_{FA} зменшується. Таким чином, для того, щоб отримати оптимальні значення ймовірностей, потрібно обирати такі значення, які більші за 13 біт (так звані послідовності Баркера та Уіларда). Оптимальні послідовності синхронізуючих слів знайдені Ньюманом, Хофманом, Морі-Стайлзом, Ліндером.

2.2.6 Мережева синхронізація

В сучасних реалізаціях мереж, де використовуються різні види модуляції, або де реалізовано односторонній зв'язок (broadcast канали), вимагається швидка

передача даних і включення кодових слів або маркерів для синхронізації знижує цю саму швидкість, тому необхідно перекласти повну відповідальність щодо синхронізації на приймача [15].

Наприклад, в системах де реалізовано підхід TDMA, параметри кінцевих прийомних пристроїв одразу повинні бути налаштовані під синхронізацію із центральним вузлом. Як відомо, в системах TDMA під кожного користувача мережі виділяється певний проміжок часу, під час якого здійснюється передача. В цьому випадку кінцевий пристрій (споживач) повинен синхронізуватись із системою, щоб усі пакети були передані виключно в момент часу, коли передача дозволена центральним вузлом.

У випадках систем із FDMA також використовується синхронізація на приймачах користувачів. Це дає можливість центральному вузлу використовуватись певний набір фільтрів для каналів та еталонний час для передачі усіма каналами. У протилежному випадку, центральний вузол буде захоплювати та супроводжувати тривалість і частоти кожного вхідного сигналу. Таким чином, синхронізація з боку приймача при “вшитих” параметрах синхронізації вирішує багато проблем.

Мережева синхронізація в приймачах поділяється в свою чергу на:

- відкриту або без зворотного зв'язку – сюди входять методи, які не залежать від параметрів сигналу на центральному вузлі (TDMA, FDMA). Фактично, кінцеві пристрої уже знають як регулювати передачу даних, в деяких випадках параметри передачі можуть модифікуватись у випадку зміни параметрів каналу передачі. Відкриті методи синхронізації дуже сильно залежать від точності та передбачуваності параметрів каналів передачі, тобто, кінцеві пристрої мають жорстку прив'язку до середовища обміну даними. Таким чином, можна зробити висновок, що методи відкритої синхронізації, краще всього використовувати у мережах, де архітектура стала та залишається незмінною. До основних переваг даного типу синхронізації відноситься: достатньо швидка синхронізація і висока

швидкість обчислень в режимі реального часу. До недоліків відносять: обов'язкова наявність “бази знань” - центрального вузла з відомими параметрами середовища передачі даних;

- закриту або зі зворотним зв'язком - практично повна протилежність відкритим методам як в перевагах, так і в недоліках. Основна відмінність полягає в тому, що для закритої синхронізації необхідний окремий канал зв'язку. Завдяки цьому, відпадає необхідність у “базі знань”, а окремий канал зв'язку дає можливість швидко та легко адаптуватись під зміну архітектури мережі або параметрів каналів передачі даних.

2.2.7 Синхронізація приймача в мережах Ethernet

В сучасному світі переважна більшість архітектур мереж реалізовано з використанням стандарту Ethernet в різних його варіаціях.

2.2.7.1 Fast Ethernet

В даному розділі розглянуті наступні стандарти Fast Ethernet на основі оптоволокна та мідної виті пари: 100Base-FX, 100Base-TX, 100Base-T4.

Слід почати з фізичного кодування сигналів для кожного стандарту, оскільки це відіграє основну роль в синхронізації приймачів. Наприклад, для стандарту 100Base-FX дані кодуються потенційними кодом NRZI, що розглянуто в розділі 2.2.3, але при цьому дані перед кодування оброблюються з допомогою кодування 4B/5B. Якщо коротко, то даний вид кодування заміняє кожні 4 біти послідовності рівня MAC на послідовність довжиною в 5 бітів. Збитковий код дозволяє уникнути ситуації створення постійного сигналу в каналі передачі, що в свою чергу дозволяє синхронізуватись приймачу без проблем. Необхідність використання збиткового коду продиктована низькою самосинхронізуючою здатність потенційних кодів, оскільки при передачі довгих послідовностей бітів часто і протягом довгого часу, сигнал може не змінюватись, що свою чергу не дає можливості визначити приймачу початок передачі даних.

Таким чином, можна зробити висновок, що використання збиткового коду вирішує проблему передачі довгих послідовностей бітів та проблему синхронізації приймача. Фактично, при використанні 5-ти бітів для кодування шістнадцяти вхідних комбінацій чотирьох бітів, можна створити таблицю кодування, в якій будь-який чотирьохбітний код буде представлений у вигляді п'ятибітової послідовності з почерговою зміною нулів та одиниць.

Стандарт 100Base-TX використовує аналогічний принцип для кодування даних, що і стандарт 100Base-FX, але задіяний інший самосинхронізуючий код, а саме MLT3. В даному випадку перевага перед NRZ/NRZI полягає в тому, що синхронізацію приймача та передавача можна виконати з допомогою послідовності логічних нулів.

Натомість стандарт 100Base-T4 використовує логічне кодування 8B/6T, в якому послідовність із восьми бітів представляється шестизначними трійковими символами.

2.2.7.2 Gigabit Ethernet

В даному розділі розглянуто принципи синхронізації приймачів для стандарту – 1000Base-T. В даних стандартах вище описані варіанти використання самосинхронізуючих кодів не працюватимуть в стандарті Gigabit Ethernet. Наприклад, Манчестерське кодування, що використовується в 10Base-T, має дуже гарні самосинхронізуючі властивості, але максимальна частота основної гармоніки чисельно дорівнює швидкості передачі даних, тобто кількості переданих біт в секунду. Цього достатньо саме для передачі даних зі швидкістю 10 Мбіт/с. Натомість, Манчестерське кодування вже не годиться для передачі даних зі швидкістю 100 Мбіт/с та вище.

В свою чергу, використання NRZI після додаткової обробки з допомогою збиткового коду 4B/5B та скремблювання, а також використання в окремих стандартах Fast Ethernet коду MLT-3 дозволяє досягти швидкості в 100 Мбіт/с та безпроблемної синхронізації приймача. Але і такий варіант не буде працювати в

даному стандарті Ethernet через фізичні властивості каналів передачі даних та самого коду.

Враховуючи усі пропозиції, стандарт 100Base-T пропонує принципово інший спосіб кодування даних – PAM-5. Даний код відображає дані з використанням п'яти фіксованих рівнів $\{-2, -1, 0, +1, +2\}$. Також, при використанні даного коду відпадає необхідність у використанні додаткового службового поля – преамбули, оскільки функція синхронізації приймача переходить на потік Idle (простий). В стандарті Gigabit Ethernet синхронізація повинна бути абсолютною та односторонньою – джерело синхронізуючих імпульсів є тільки один компонент, який використовує свої імпульси для синхронізації сигналу. Використання імпульсів та годинника для синхронізації у випадку Gigabit Ethernet виправдано, оскільки така швидкість досягається використанням чотирьох ліній для передачі, де в кожній лінії досягається швидкість 250 Мбіт/с. Даний підхід називається протоколом PTP (Precision Time Protocol).

3 МОДЕЛЮВАННЯ ТА РЕАЛІЗАЦІЯ

3.1 Математичне моделювання процесів синхронізації

Важливим етапом розробки кореляційного приймача є розуміння принципів синхронізації на фізичному рівні у математичному вигляді. В даному розділі буде розглянуто схеми синхронізації на основі контуру ФАПЧ, АКФ та ВКФ.

3.1.1 Частотна та фазова синхронізація

В більшості схем синхронізації завжди наявні різновиди контуру ФАПЧ. В сучасних приймачах знайти контур синхронізації достатньо складно, або зовсім неможливо, але функціональний аналог присутній в будь-якому випадку. Структурна схема контуру ФАПЧ зображено на рисунку 2.3.

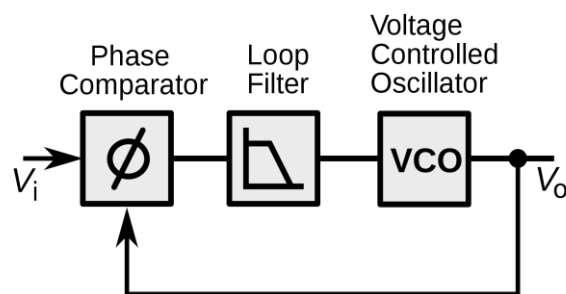


Рисунок 20 – Структурна схема
ФАПЧ

Контур ФАПЧ саморегулюється та самоконтролюється, а регулюючим параметром є фаза, згенерованого локально, сигналу, який в свою чергу є копією несучого сигналу, що попадає на приймач. З рисунку 3.24 видно, що схема ФАПЧ складається з 3-х елементів:

- фазовий порівнювач, або детектор фази;
- фільтр контуру;

— генератор, керований напругою – (ГКН). В цифрових схемах використовуються системний годинник або лічильник-дільний під цифровим керуванням.

Фазовий детектор – це спеціалізований пристрій, який займається вимірюванням різниці фаз між локально згенерованим сигналом (опорним) та сигналом, що поступає на вхід приймача. В результаті, фазовий детектор визначає різницю фаз, у випадку коли порівнювані сигнали змінюються один відносно іншого, яку передає на контурний фільтр. При проходженні сигналу через фільтр контуру, виділяється сигнал певних частот, і далі використовується керуючим генератором (ГКН) в якості керуючого елементу.

Для сучасних цифрових приймачів схема ФАПЧ може бути більш складною в математичному плані. Наприклад, детектор фази може представляти собою набір кореляторів або узгоджених фільтрів, кожен з яких здійснює порівняння зі своїм опорним сигналом і з-поміж усіх порівнянь обирається найбільш схожий опорний сигнал, а потім відповідно знаходиться різниця фаз.

Крім того, ГКН на завжди може представляти з себе генератор синусоїдального сигналу. Це може бути постійна пам'ять, в якій вказівники керуються таймерами і пристроєм, який визначає різницю фаз.

Зворотній зв'язок, що представлений контуром на рисунку 3.24 не завжди може бути безперервним, а зміна фази може відбуватись в певні часові періоди, коли приходить кадр або пакет даних. Все залежить від структури сигналу, який подається на вхід приймача.

Для більшого розуміння роботи контуру ФАПЧ слід розглянути приклад. Для початку слід взяти вхідний сигнал, який описується наступним рівнянням:

$$r(t) = \cos[\omega_0 t + \theta(t)]. \quad (4)$$

У рівнянні **Ошибка! Источник ссылки не найден.** ω_0 - частота несучого сигналу, $\theta(t)$ – фаза, яка повільно змінюється. Опорний сигнал ГКН описується наступним рівнянням:

$$x(t) = -2\sin[\omega_0 t + \hat{\theta}(t)]. \quad (5)$$

Відповідно, за рахунок зворотнього зв'язку у детектора фази, на виході того ж детектора фази будуть отримати сигнал, як описується наступним рівнянням:

$$\begin{aligned} e(t) = x(t)r(t) &= 2\sin[\omega_0 t + \hat{\theta}(t)]\cos[\omega_0 t + \theta(t)] = \\ &= \sin[\theta(t) - \hat{\theta}(t)] + \sin[2\omega_0 t + \theta(t) + \hat{\theta}(t)] \end{aligned} \quad (6)$$

Прийmemo, що в контурі ФАПЧ фільтр є фільтром нижнім частот, як вказувалось вище (оскільки зазвичай, практична реалізація контуру передбачає наявність саме такого фільтра). В такому випадку, у рівнянні **Ошибка! Источник ссылки не найден.** в сумі другим доданком можна знехтувати і в результаті залишиться синусоїда різниці фаз вхідного сигналу **Ошибка! Источник ссылки не найден.** та опорного сигналу ГКН **Ошибка! Источник ссылки не найден.** З вигляду залишку впливає, що це частина, яка необхідна для визначення різниці фаз. Якщо зробити припущення, що ω_0 некерована частота ГКН (частота на виході при нульовій напрузі), то різниця між вихідною частотою ГКН від ω_0 виражається як похідна по часу від фази $\hat{\theta}(t)$. Вихідна частота на виході ГКН лінійно залежить від вхідної напруги. Таким чином, оскільки вихідна нульова напруга дає частоту ω_0 , то різниця різницю частоти можна виразити наступним рівнянням:

$$\begin{aligned} \Delta\omega(t) &= \frac{d}{dt}[\hat{\theta}(t)] = K_0 y(t) = \\ &= K_0 y(t) * f(t) = K_0[\theta(t) - \hat{\theta}(t)] * f(t) \end{aligned} \quad (7)$$

У формулі **Ошибка! Источник ссылки не найден.** $\Delta\omega(t)$ – різниця частот, а * - операція згортки. Оскільки в даному прикладі розглядаються приближення малих кутів, то справедлива наступна рівність $\sin[\theta(t) - \hat{\theta}(t)] = \theta(t) - \hat{\theta}(t)$. Приближення малих кутів працює тільки при малих значеннях вихідного неузгодження за фазою. Такі умови виконується лише, коли ФАПЧ контур нормально функціонує. Із формулі **Ошибка! Источник ссылки не найден.** K_0 – коефіцієнт підсилення ГКН, а $f(t)$ – імпульсна характеристика фільтру (в даному випадку – фільтру низьких частот). Диференціальне рівняння виду **Ошибка! Источник ссылки не найден.** відносно $\hat{\theta}(t)$ називається лінеаризованим рівнянням контуру.

3.2 Математичне моделювання кореляційних функцій

3.2.1 Автокореляційна функція

Автокореляційна функція (АКФ) використовується для визначення рівня відмінності сигналу від його зміщеної копії. Фактично, АКФ це визначення кореляції сигналу із самим собою в різні точки часу **Ошибка! Источник ссылки не найден.** АКФ визначається за наступною формулою:

$$B_u(\tau) = \int_{-\infty}^{\infty} u(t)u(t - \tau)dt. \quad (8)$$

Далі наведено список властивостей АКФ:

- парність – $B_u(\tau) = B_u(-\tau)$;
- при будь-якому значення зміщення по часу, значення модулю АКФ не стає більшим за енергію сигналу: $|B_u(\tau)| \leq B_u(0) = E_u$;
- автокореляція для періодичної функції – періодична функція з тим же періодом;
- автокореляція суми двох некорелюючих функцій – це сума автокореляцій цих функцій;

— автокореляція досягає свого максимуму в точці 0 на осі абсцис;

АКФ є симетричною функцією, в якій існує центральний максимум, який в свою чергу завжди додатній. Крім того, в залежності від виду сигналу АКФ може приймати вигляд або монотонно спадаючої, або коливальної функції. На рисунку 3.24 зображено графіки прямокутного сигналу та графік АКФ, а на рисунку 3.24 АКФ сигналу типу “пилка”.

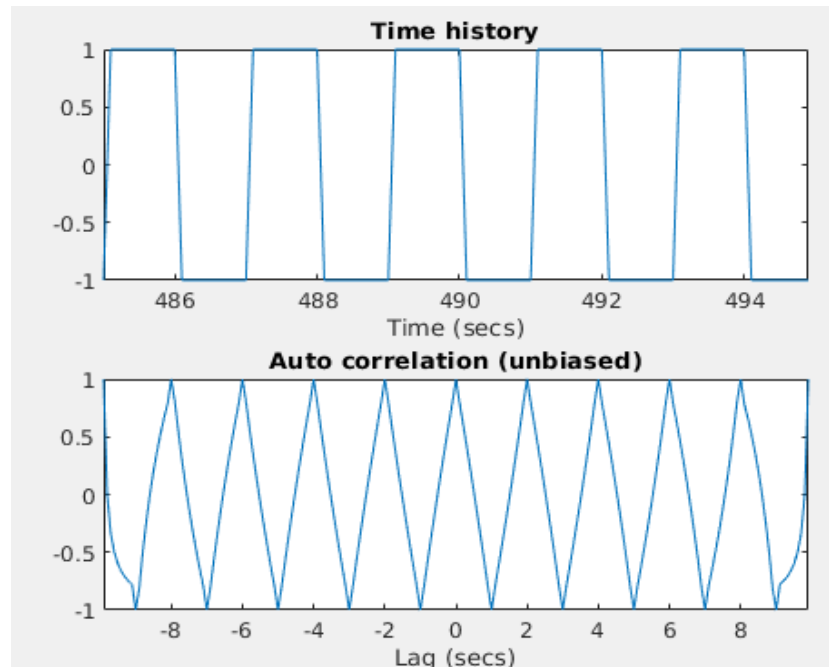


Рисунок 21 – Вигляд АКФ для прямокутного сигналу згенерованого засобами MatLab Simulink

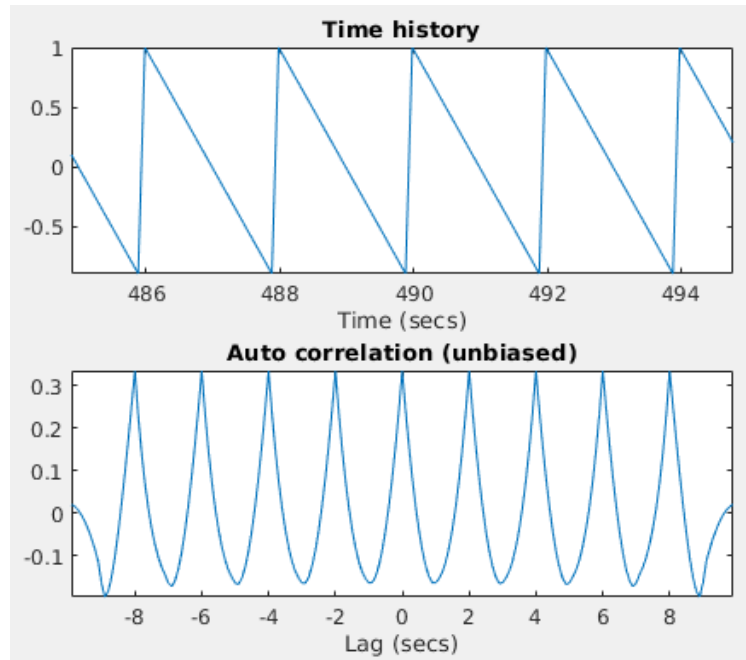


Рисунок 22 – Вигляд АКФ для сигналу вигляду “пила”
згенерованого засобами MatLab Simulink

Прямокутний сигнал з рисунку 3.24 будується як інтервал, що поділений на ціле число проміжків або позицій N , де $N > 1$. Відповідно, в одній із таких позицій сигнал може отримувати значення $+1$ або -1 , що відповідає логічній одиниці та логічному нулю. Розглянемо складний сигнал на рисунку 3.24, де N становить 4.

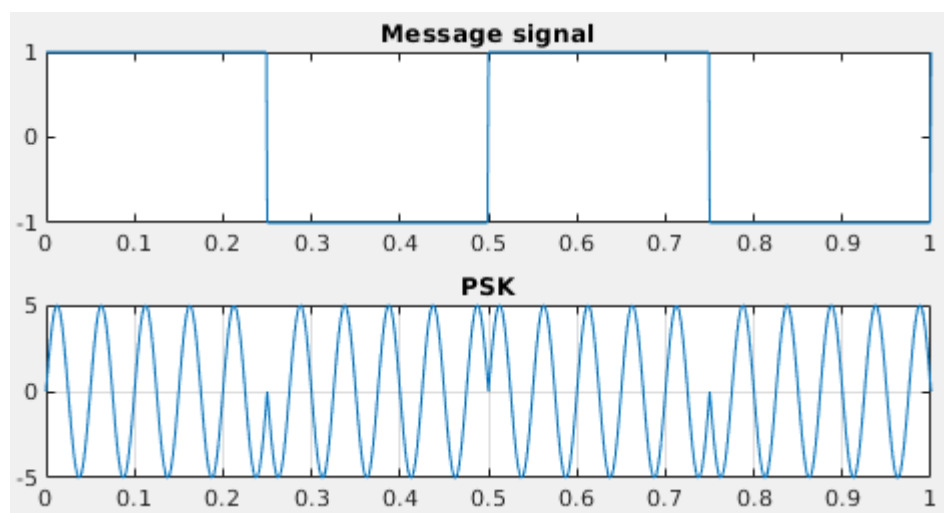


Рисунок 23 – Чотирьохпозиційний складний сигнал

Можна помітити, що вигляд дискретного сигналу може бути різним. У випадку прямого представлення інформаційного сигналу на рисунку 2.4 (верхній графік), то значенню +1 відповідає додатне значення U висоти прямокутного імпульсу, а -1 – відповідно від’ємне значення U . У випадку фазового кодування, що зображено на нижньому графіку рисунку 2.4 (PSK), для передачі символу +1 на відповідних позиціях генеруються відрізки гармонічного сигналу з нульовою початковою фазою. Для передачі символу -1, використовується відрізок синусоїди такої ж довжини і частоти, але при цьому сдвиг фази складає 180 градусів. При усій зовнішній різниці обох сигналів, між ними можна встановити певну схожість, якщо звернутись до їхніх математичних моделей, а саме: кожен цей сигнал є представленням ряду чисел $\{a_1, a_2, \dots, a_{N-1}, a_N\}$, де кожен символ a приймає одне з двох значень – +1 або -1. Для подальшого розуміння АКФ слід звернутись до прикладу, де буде розглянуто сигнал, який буде представлено у вигляді ряду 4 значень - наприклад, $\{1, -1, 1, -1\}$. Для роботи з цим рядом, слід використати розгорнуту форму запису дискретного сигналу – $\{...0001 - 11 - 100...\}$.

При обробці дискретних сигналів в контексті АКФ, необхідно здійснювати зсув оригінального сигналу на декілька позицій в бік запізнення без зміни його форми, наприклад:

$\{...00011110000...\}$

$\{...00001111000...\}$

$\{...00000111100...\}$

$\{...00000011110...\}$

і тд.

Для використання формули **Ошибка! Источник ссылки не найден.** необхідно зробити перетворення, щоб став можливим обрахунок дискретного аналогу АКФ. В даному випадку операцію інтегрування слід замінити сумою, а τ замінити на n , що означає кількість позицій зсуву копії оригінального сигналу.

Після даного перетворення формула **Ошибка! Источник ссылки не найден.** набуває наступного вигляду:

$$\hat{B}_u(n) = \sum_{j=-\infty}^{\infty} u_j u_{j-n}. \quad (9)$$

Всі властивості АКФ повністю зберігаються в її дискретному аналозі. Для наочного зображення роботи дискретної АКФ слід розрахувати значення для чотирьохпозиційного сигналу, який представляє з себе наступну послідовність: {1,1,1,1}. Далі наведено перелік послідовностей із зсувами:

{...00011110000...}

{...00001111000...}

{...00000111100...}

{...00000011110...}

{...00000001111...}

Можна помітити, що при $n = 4$ оригінальний сигнал та його зсунута копія вже не накладаються один на одного, а тому результат суми за формулою **Ошибка! Источник ссылки не найден.** буде дорівнювати 0. Для прикладу, далі наведено аналітичний розрахунок АКФ за формулою **Ошибка! Источник ссылки не найден.**:

$$\hat{B}_u(0) = 1 + 1 + 1 + 1 = 4$$

$$\hat{B}_u(1) = 1 + 1 + 1 = 3$$

$$\hat{B}_u(1) = 1 + 1 = 2$$

$$\hat{B}_u(1) = 1$$

Результати аналітичного розрахунку зображено на рисунку 2.5 у вигляді пелюсток АКФ.

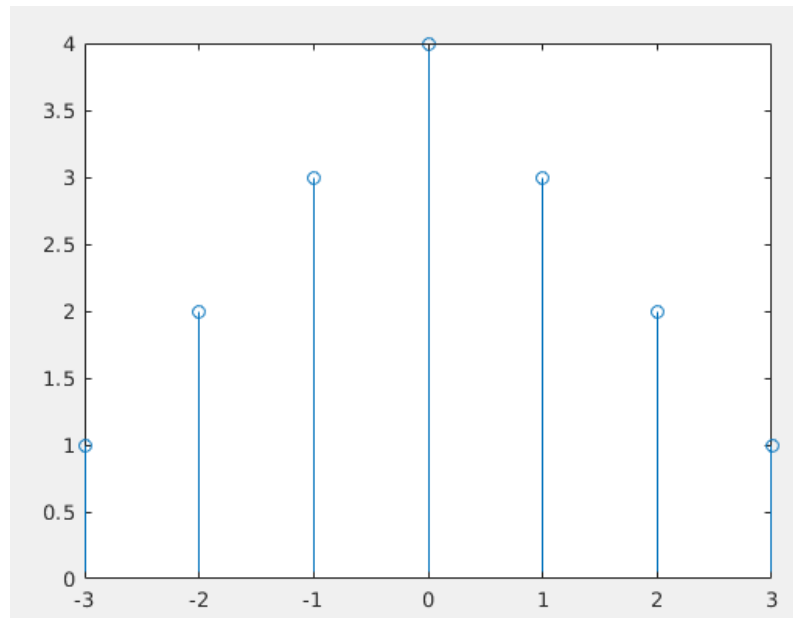


Рисунок 24– Пелюстки АКФ для послідовності $\{1,1,1,1\}$

З графіку на рисунку 3.24 можна помітити, що бокові пелюстки АКФ лінійно співпадають з ростом номеру n . Схожу картину можна спостерігати при обчисленні АКФ на аналоговому прямокутному сигналі.

Для подальшого порівняння результатів АКФ слід обрати іншу послідовність – $\{1, -1, 1, -1\}$, що відрізняється від попередньої інвертованими двома елементами. Результатом обчислення за формулою **Ошибка! Источник ссылки не найден.** є графік АКФ на рисунку 3.24 де можна спостерігати наступну залежність: перший боковий пелюсток змінює свій знак, але абсолютне значення залишається незмінним.

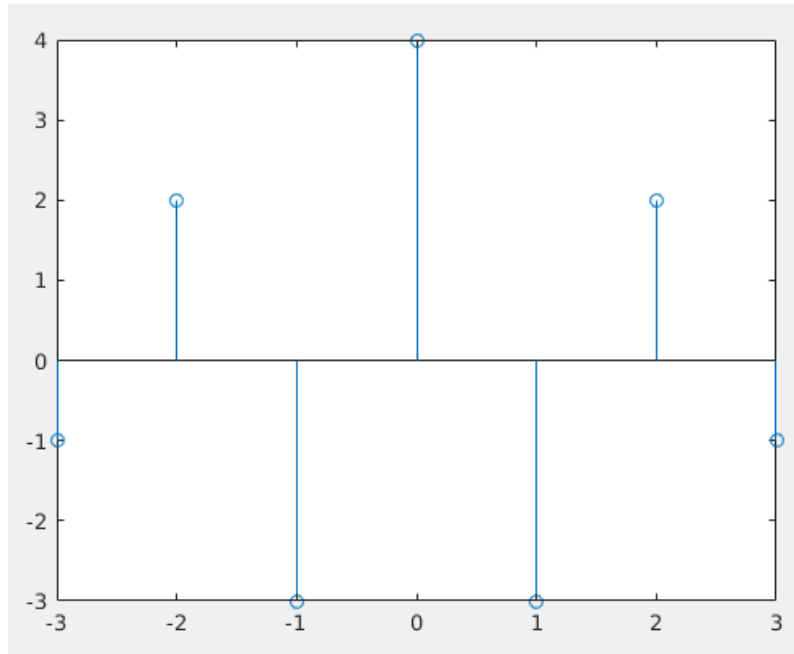


Рисунок 25 – Пелюстки АКФ для послідовності $\{1, -1, 1, -1\}$

Для даного набору чисел існує тільки два варіанти послідовності, які найкраще проявлятимуть власні кореляційні властивості, а саме: $\{1, 1, 1, -1\}$, $\{1, 1, -1, 1\}$. На рисунку 3.24 відповідно зображено пелюстки АКФ кожної послідовності для порівняння.

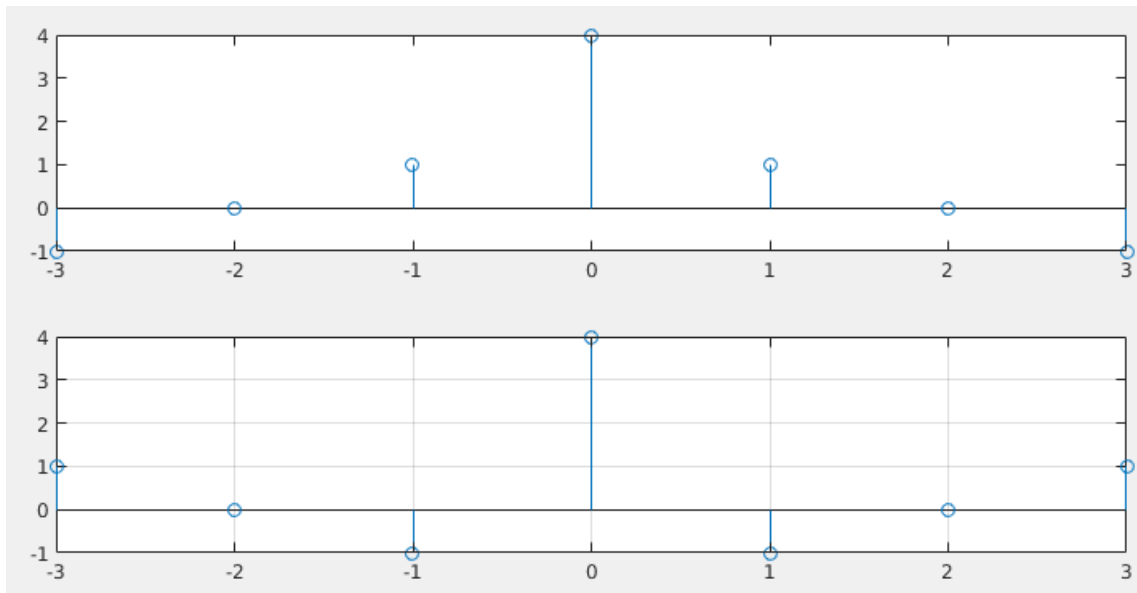


Рисунок 26 – Пелюстки АКФ для послідовностей $\{1, 1, 1, -1\}$, $\{1, 1, -1, 1\}$

Також, послідовності, що зображені на рисунку 3.24 включені в список так званих сигналів Баркера – список сигналів з найбільш ідеальними кореляційними властивостями.

3.2.2 Взаємнокореляційна функція (ВКФ) або взаємна кореляція двох сигналів.

ВКФ дозволяє єдиним чином описати як відмінність у формі сигналів, так і їх взаємне розташування на осі часу. Використовуючи формулу **Ошибка!** **Источник ссылки не найден.** опишемо формулу для ВКФ:

$$B_{uv}(\tau) = \int_{-\infty}^{\infty} u(t)v(t - \tau)dt, \quad (10)$$

де $u(t)$ та $v(t)$ – деякі сигнали представлені у вигляді числових послідовностей.

Для більшого розуміння доцільності використання ВКФ слід розглянути випадок, коли значення інтегрального виразу **Ошибка! Источник ссылки не найден.** дорівнює 0, тобто сигнали знаходяться в ортогональному стані. У

випадку, коли сигнал $v(t)$ проходить через певну кількість пристроїв, він буде зміщений відносно $u(t)$ на деякий час τ . Таким чином, можна стверджувати, що взаємкореляційна функція є характеристикою, що визначає стійкість ортогонального стану зсунутих сигналів.

Далі наведено властивості ВКФ:

- ВКФ не є парною та не є непарною функцією;
- якщо випадкові сигнали $u(t)$ та $v(t)$ генеруються незалежними джерелами і не містять в собі періодичності, то для них значення ВКФ прямує до 0 і ці функції не мають ніякого зв'язку.

На рисунку 3.24 зображено приклад ВКФ для прямокутного та трикутного імпульсів.

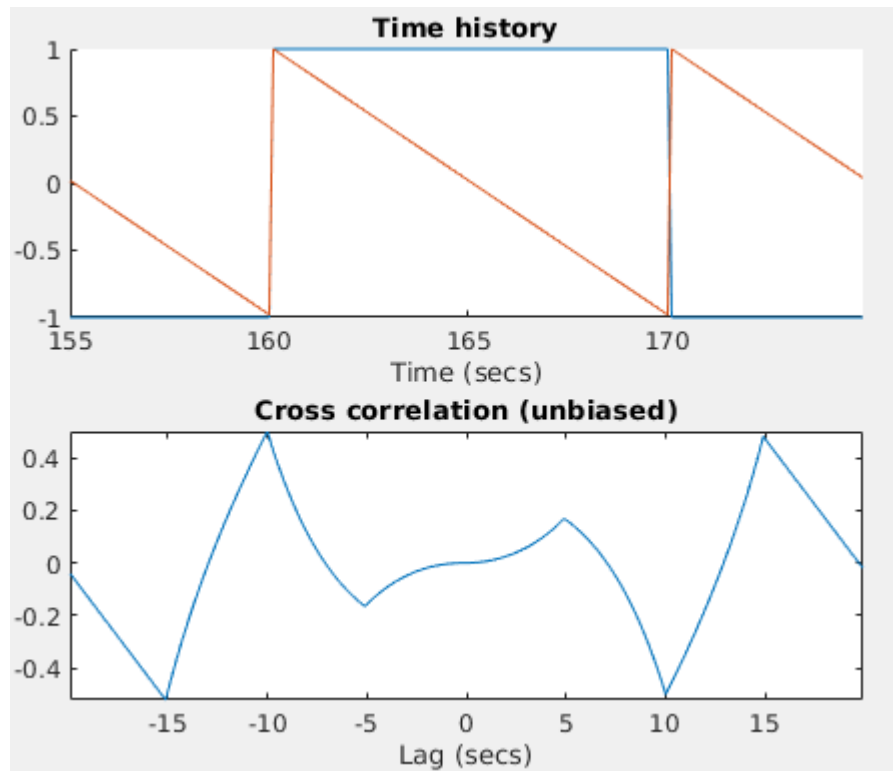


Рисунок 27 – Графік ВКФ (нижній) та сигналів
(верхній)

Для використання ВКФ у випадку дискретних сигналів необхідно піти таким самим шляхом як і у випадку автокореляції. Сигнали з однаковими періодами задані наступними числовими рядами:

$$u = \{\dots, u_{-1}, u_0, u_1, u_2, \dots\},$$

$$v = \{\dots, v_{-1}, v_0, v_1, v_2, \dots\}.$$

Аналогічно як в АКФ формула для ВКФ матиме наступний вигляд:

$$B_{uv}^{\wedge}(n) = \sum_{j=-\infty}^{\infty} u_j v_{j-n}, \quad (11)$$

де n – число в множині цілих чисел.

Для більш наочного розуміння роботи ВКФ слід розглянути попередній приклад з послідовністю з чотирьох елементів, але в даному випадку взяти дві послідовності, а саме: $\{1, 1, 1, -1\}$, $\{1, 1, -1, 1\}$. Подібно алгоритму для АКФ зміщуємо другу послідовність в бік запізнення. Отримаємо наступний результат:

$$\begin{aligned} &\{...0000111 - 100000...\} \\ &\{...000011 - 1100000...\} \\ &\{...0000011 - 110000...\} \\ &\{...00000011 - 11000...\} \\ &\{...000000011 - 1100...\} \end{aligned}$$

За формулою **Ошибка! Источник ссылки не найден.** обраховуємо значення ВКФ і отримаємо наступні результати:

$$B_{uv}^{\wedge}(0) = 0, B_{uv}^{\wedge}(1) = 3, B_{uv}^{\wedge}(2) = 0, B_{uv}^{\wedge}(3) = -1.$$

Таким самим чином формуємо таблицю зі зсувом в бік випередження:

$$\begin{aligned} &\{...0000111 - 1000...\} \\ &\{...000011 - 11000...\} \\ &\{...00011 - 110000...\} \\ &\{...0011 - 1100000...\} \\ &\{...011 - 11000000...\} \\ &B_{uv}^{\wedge}(-1) = 1, B_{uv}^{\wedge}(-2) = 0, B_{uv}^{\wedge}(-3) = 1. \end{aligned}$$

Також, на рисунку 3.24 представлена діаграма, яка відображає графік ВКФ цих двох сигналів.

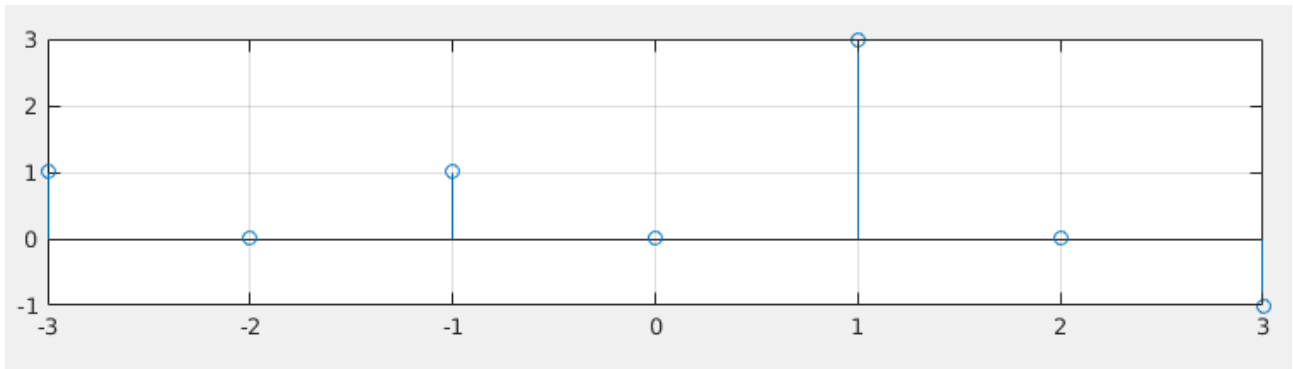


Рисунок 28 – Графік ВКФ для послідовностей $\{1,1,1,-1\}$ та $\{1,1,-1,1\}$

3.3 Моделювання АКФ та ВКФ в середовищі MatLab/Simulink

Для розробки алгоритму синхронізації на основі АКФ та ВКФ для синхронного приймача необхідно здійснити попереднє математичне моделювання засобами MatLab/Simulink **Ошибка! Источник ссылки не найден..** Це дасть в свою чергу більше розуміння, яким чином необхідно здійснювати пошук схожих сигналів, зміщення одного відносно іншого тощо.

Для базового розуміння моделі роботи ВКФ для знаходження запізнення у цифрових сигналах слід звернутись до просто прикладу, в якому наявний простий сигнал та його зміщена копія з накладеним шумом. Для прикладу, копія сигналу зміщена на 3 елементи. Даний сигнал зображено на рисунку 3.24. ВКФ буде використана для знаходження запізнення в копії. Всі операції будуть проводитись з дискретними сигналами.

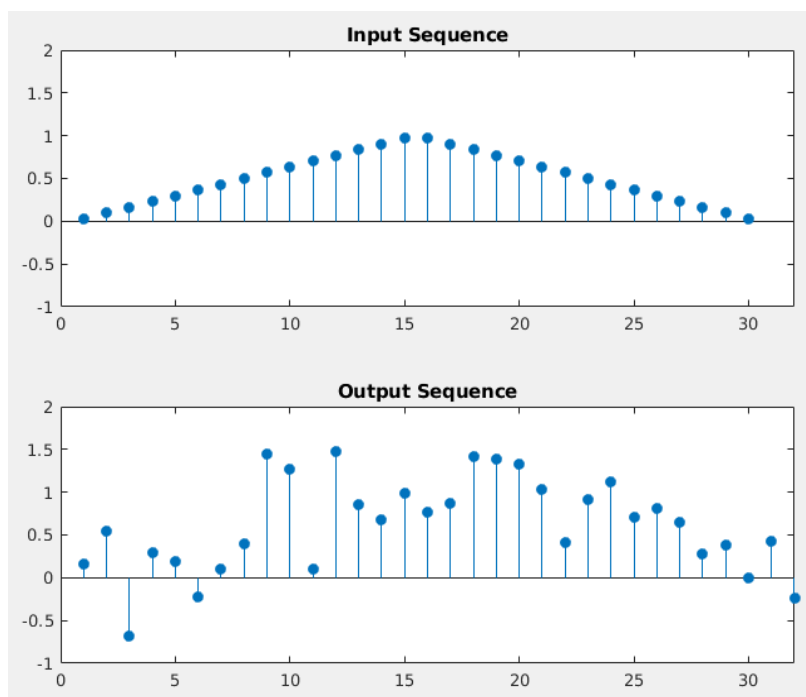


Рисунок 29 – Приклад функцій трикутної та її зміщеної копії із шумом

Тепер необхідно обрахувати значення ВКФ та відобразити графік, який можна побачити на рисунку 3.24.

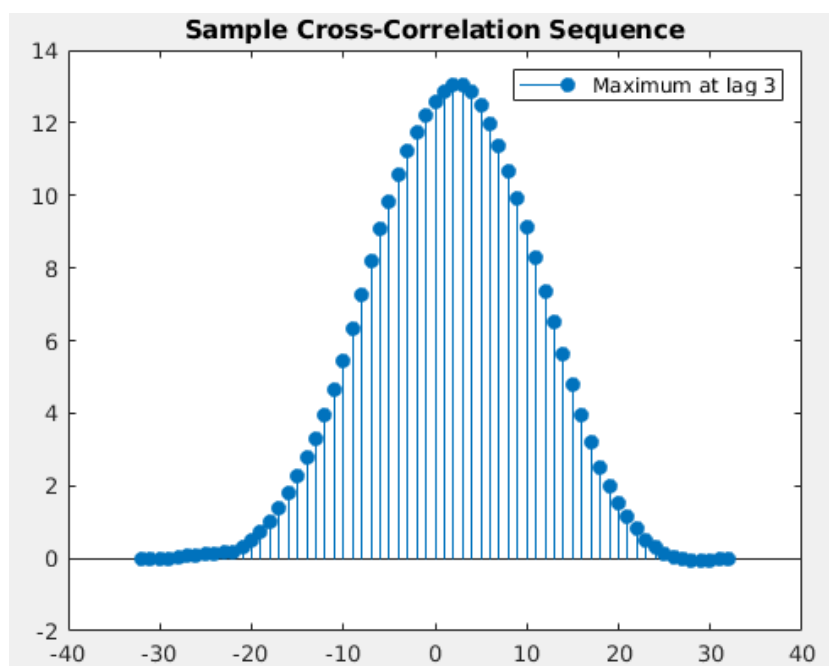


Рисунок 30 – Графік ВКФ для прикладу

В наступному прикладі використані сигнали більшої довжини та більш зашумлені. На рисунку 3.24 зображено графіки сигналів, які підписані відповідним чином: “Signal example 1”, “Signal example 2”, “Signal”. Перші два – це приклади сигналів де буде здійснюватись пошук характеристики для сигналу “Signal”.

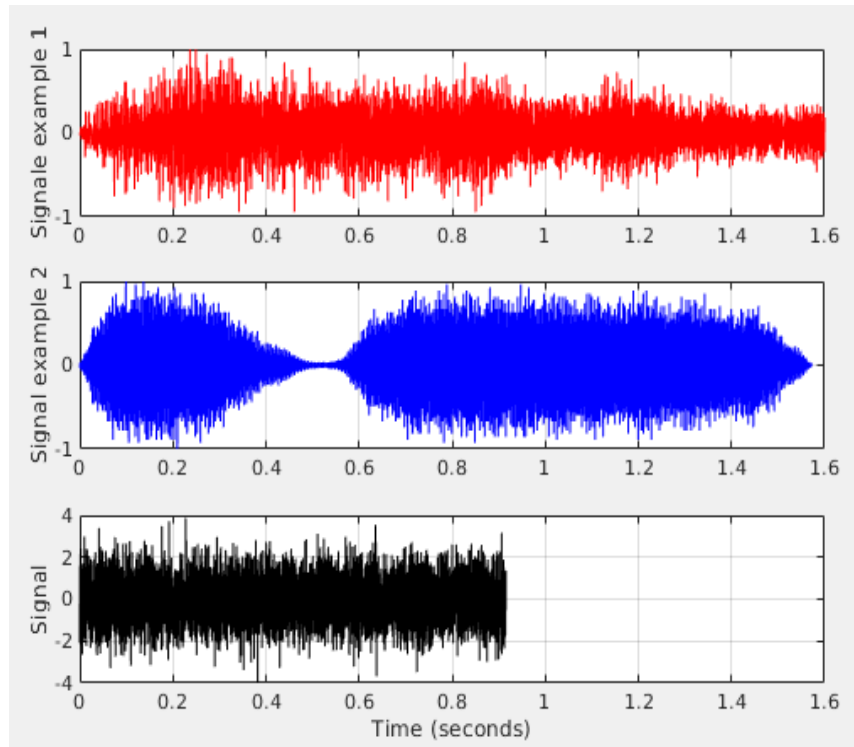


Рисунок 31 – Приклад сигналів для моделювання кореляції

Наступним кроком буде визначення графіку ВКФ для наступного набору сигналів: “Signal example 1” та “Signal”, “Signal example 2” та “Signal”. На рисунку 3.24 зображено відповідні графіки ВКФ.

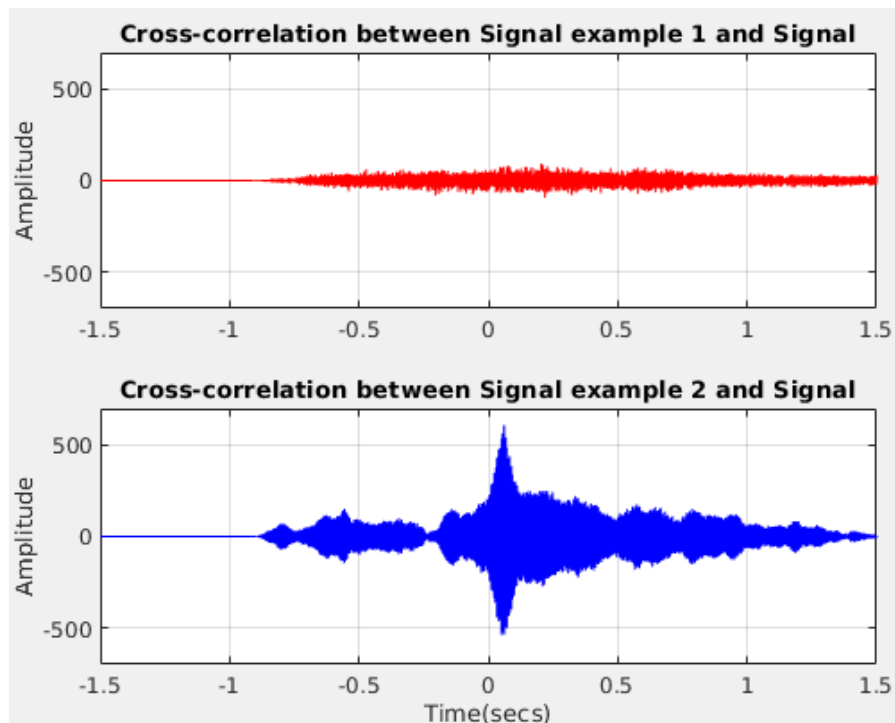


Рисунок 32 – Графіки ВКФ для набору сигналів “Signal example 1” та “Signal”, “Signal example 2” та “Signal”

З рисунку 3.24 видно наступну залежність: на першому графіку помітно, що значення кореляції “Signal” та “Signal example 1” менше ніж значення кореляції “Signal” та “Signal example 2”. Така залежність визначається наявністю максимуму на другому графіку, натомість на першому цей максимум відсутній. Відповідно, порівнюючи обидва значення, обирається найбільше, що в свою чергу відповідає другому графіку ВКФ.

Наступним кроком необхідно обрахувати час затримки: це здійснюється з допомогою обрахунку затримки у вигляді кількості елементів затримки – Samples. Після цього дуже просто обрахувати це значення у вигляді часу, поділивши на частоту дискретизації. Для даного набору сигналів після обрахунку засобами Matlab були отримані наступні значення:

- зміщення за кількістю елементів затримки – 499;
- зміщення за часом – 0.0609 секунд.

Таким чином, можна зробити висновок, що максимум графіку ВКФ вказує на те, що сигнал “Signal” знаходиться в прикладі сигналу “Signal example 2” після 0.0609 секунди.

Наступним кроком для створення моделі є обрахунок значення затримки між сигналами та їх вирівнювання. Для цієї задачі згенеровано новий випадковий послідовності сигналів, що зображені на рисунку 3.24.

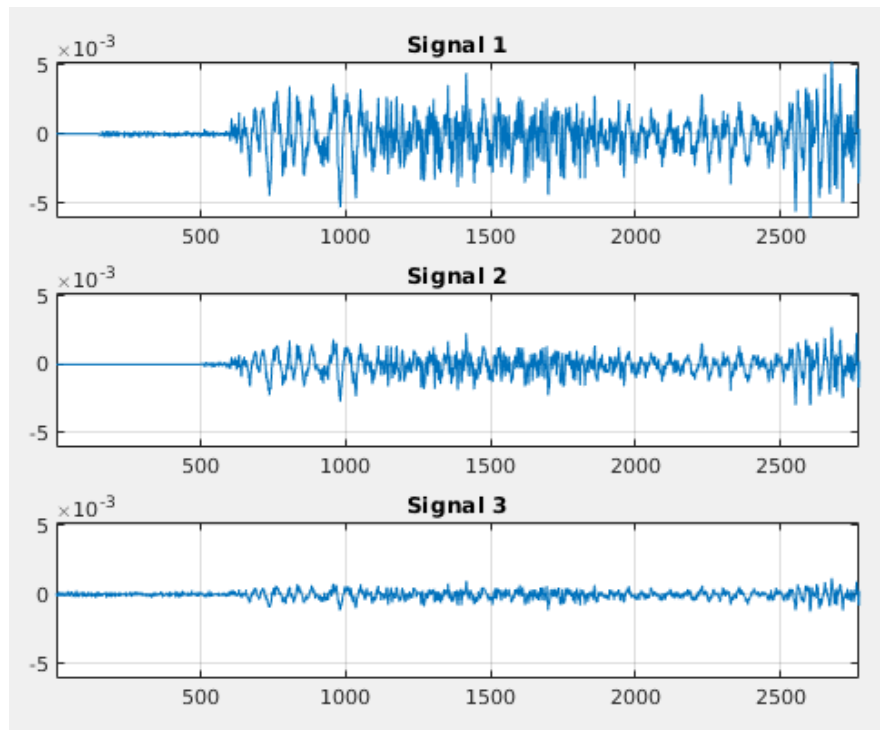


Рисунок 33 – Набір випадково згенерованих сигналів

Затримка між сигналами обраховується з використанням функції `finddelay`.
Результати виконання даної наступні:

- затримка між Signal 1 та Signal 2: -350;
- затримка між Signal 1 та Signal 3: 150.

Розрахунок затримки здійснюється за наступним алгоритмом: спочатку обраховується взаємна кореляція між парою сигналів у всіх точках запізнення. Після цього, необхідно розрахувати нормалізовану взаємну кореляцію. Очікувана

затримка знаходиться як від'ємна затримка для максимального абсолютного значення нормалізованої ВКФ.

У випадку, коли при знаходженні значення кореляційної функції отримується декілька аюсолютно найбільших значень (у випадку періодичних сигналів), то затримка визначається як від'ємне серед найменших із затримок. На рисунку 3.24 можна запевнитись у правильності проведених розрахунків. Наприклад, різниця між початком Signal 1 та Signal 2 різниця складає -350, а між Signal 2 та Signal 3 – -150. Додати на скрін відмітки

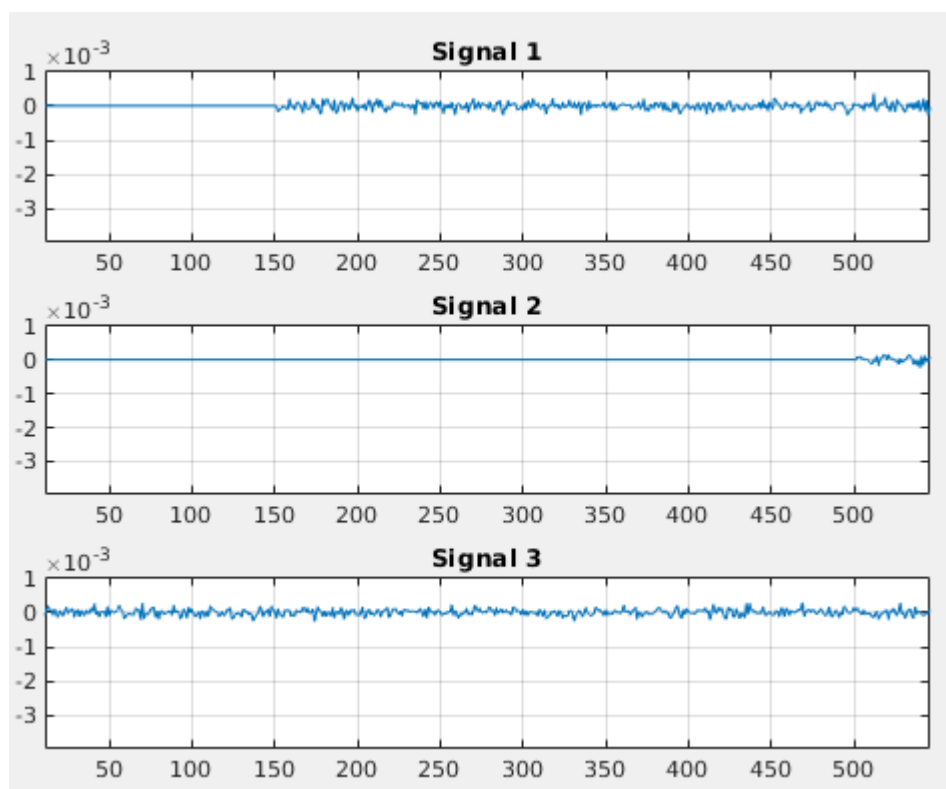


Рисунок 34 – Відмасштабовані початки сигналів

Велика кількість вимірюваних даних отримуються асинхронно з декількох джерел. У випадку, коли ці сигнали необхідно дослідити в один момент необхідно їх синхронізувати. Для цих цілей використовується ВКФ. На рисунку 3.24 зображено ще один приклад випадково згенерованих сигналів.

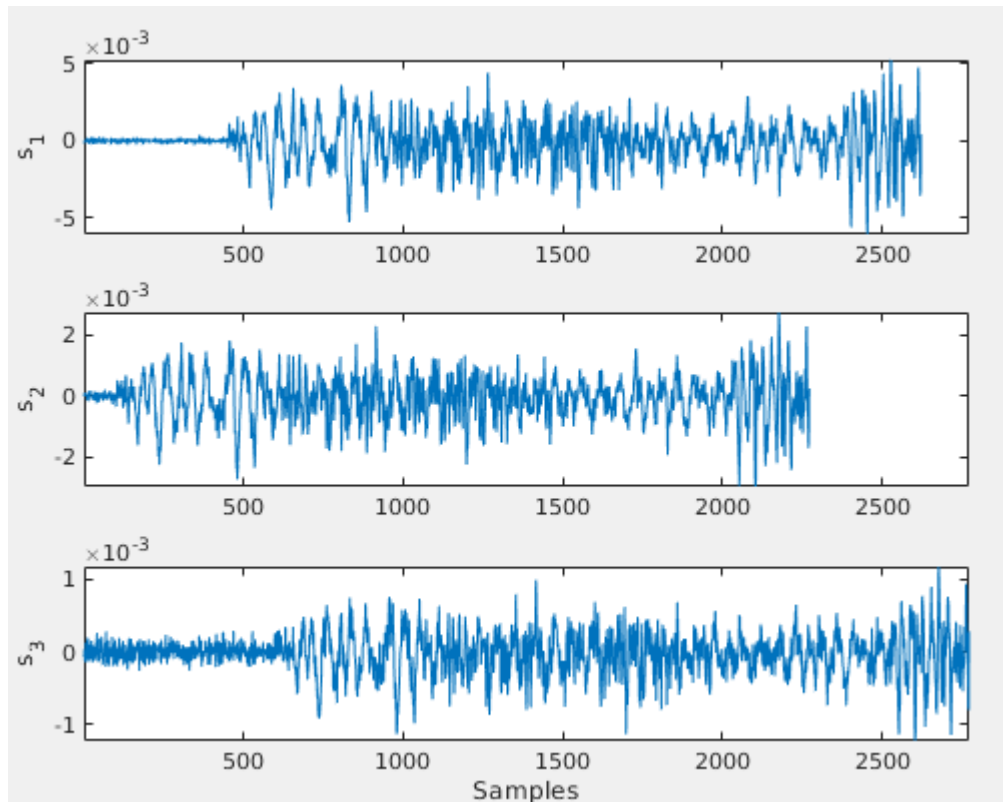


Рисунок 35 – Приклади сигналів для суміщення сигналів

Наступним кроком буде обрахунок значень ВКФ для наступних пар сигналів: S2 та S1, S3 та S1, S3 та S2. Після цього необхідно нормалізувати значення взаємної кореляції, щоб в послідовностях були лише по одному максимуму кореляції. Для більш наочного розуміння процесу обрахунку слід навести невеликий фрагмент коду із середовища MatLab:

$$[C21, lag21] = xcorr(S1, S1)$$

$$C21 = C21 / \max(C21)$$

Аналогічним чином обраховуються C31, C32 для решти пар послідовностей. Далі необхідно знайти максимальне значення функції кореляції. На рисунку 3.24 зображено графіки ВКФ для кожної пари сигналів.

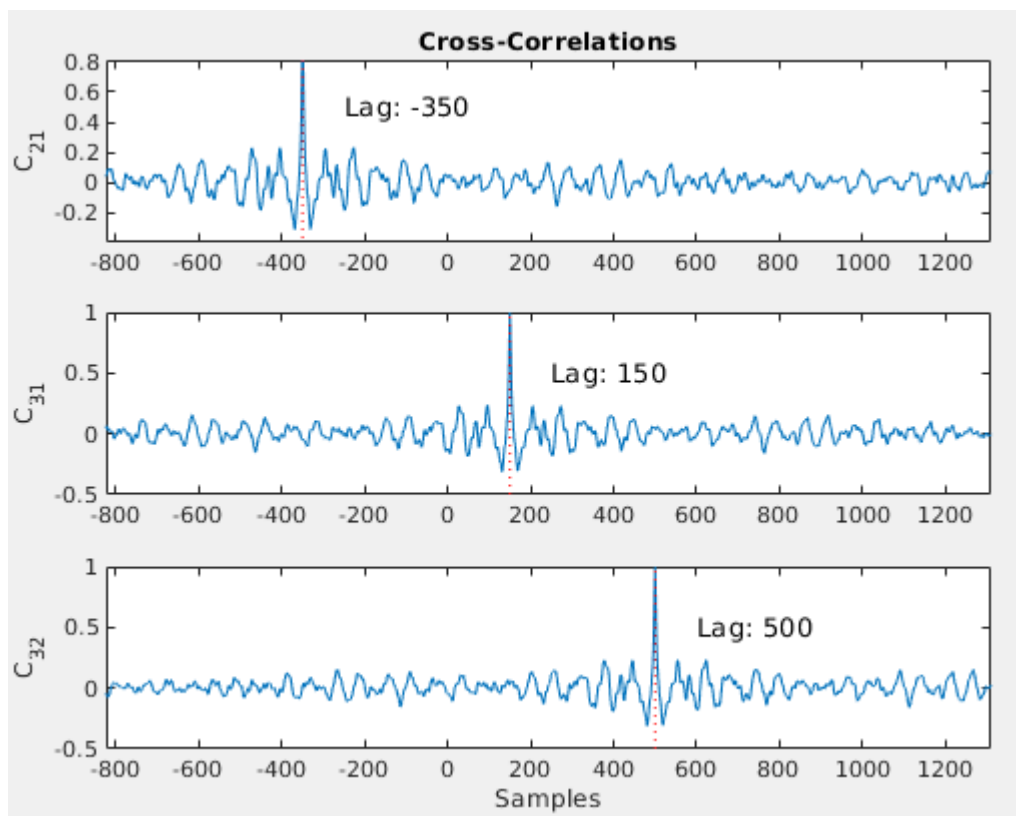


Рисунок 36 – Графіки ВКФ для кожної з пари сигналів

З рисунку 3.24 можна помітити, що S2 випереджає S1 на 350 значень, S3 запізнюється на 150 значень відносно S1, а S2 випереджає S3 на 500 значень. Для того, щоб сумістити графіки, необхідно здійснити зріз сигналів, де початковим індексом є максимальне значення ВКФ по модулю і до кінця послідовності. Після цього сигнали синхронізовані і можуть бути оброблені одночасно незалежно від зміщення.

3.4 Реалізація моделі засобами Python

В рамках дипломної дисертації було створено спеціалізований додаток, що емулює в режимі реального часу вирівнювання прийманих сигналів відносно базового з використанням ВКФ та АКФ.

3.5 Реалізація моделі приймача засобами Matlab

Перед початком програмування сигнального процесора алгоритмами пошуку максимального значення ВКФ необхідно визначити параметри цієї функції. Першим параметром є підбір послідовностей опорних сигналів. Це найважливіший етап в побудуванні кореляційного приймача, бо якщо, значення ВКФ на обраних послідовностях будуть мати великі значення, то збільшується ймовірність помилкових спрацювань щодо визначення проміжків кадрів.

Важливим етапом в розробці приймача на основі ВКФ є вибір опорних сигналів, відносно яких буде здійснюватись розрахунок кореляції. Це необхідно для того, що максимально ефективно знаходити значення зміщення в прийманому сигналі. Для більш наочного розуміння даного алгоритму, взято послідовність {1 1 1 0 1 0 1 0 1}, закодовану кодом NRZ. На рисунку 3.24 зображено графік такого

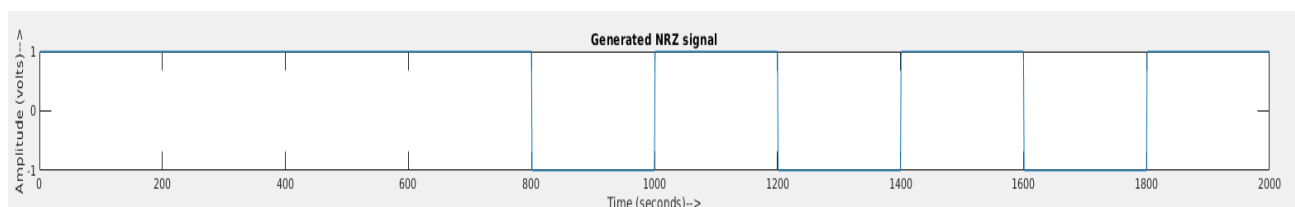


Рисунок 37 – Закодована послідовність {1 1 1 0 1 0 1 0 1} кодом NRZ

сигналу.

Далі даний сигнал модулюється BPSK модуляцією та формується сигнал, що зображено на рисунку 3.24. Якщо ставити у відповідність до графіку з рисунку 3.24 можна побачити на рисунку 3.24 фазові переходи відповідно.

Далі треба здійснити зсув на 170 секунд по осі абсцис. Після цього необхідно розрахувати значення ВКФ та відобразити графік. В якості опорного сигналу використано послідовність, що відповідає послідовності сигналу. Графік ВКФ зображено на рисунку 3.24 та відповідно розраховане випередження сигналу. Можна помітити, що значення випередження дорівнює 20 секундам, а не

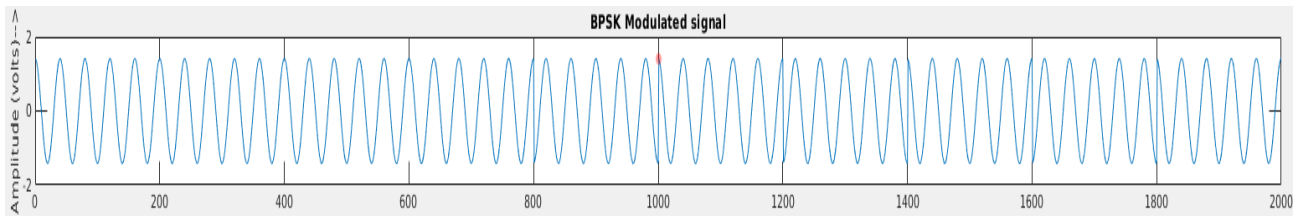


Рисунок 38 – Сигнал модульований BPSK

170 як задано. Це яскраво показує момент, що необхідно заздалегіть готувати

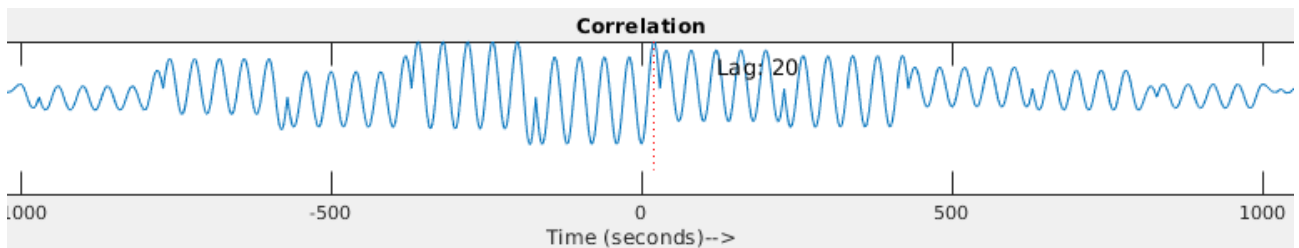


Рисунок 39 – Графік ВКФ послідовності {1 1 1 1 0 1 0 1 0 1}

набір опорних сигналів для правильного розрахунку затримки або випередження.

Після розрахунку затримки або випередження, необхідно здійснити зсув прийнятого сигналу, провести демодуляція та дістати інформаційний сигнал. Отриманий інформаційний сигнал показано на рисунку у вигляді імпульсів.

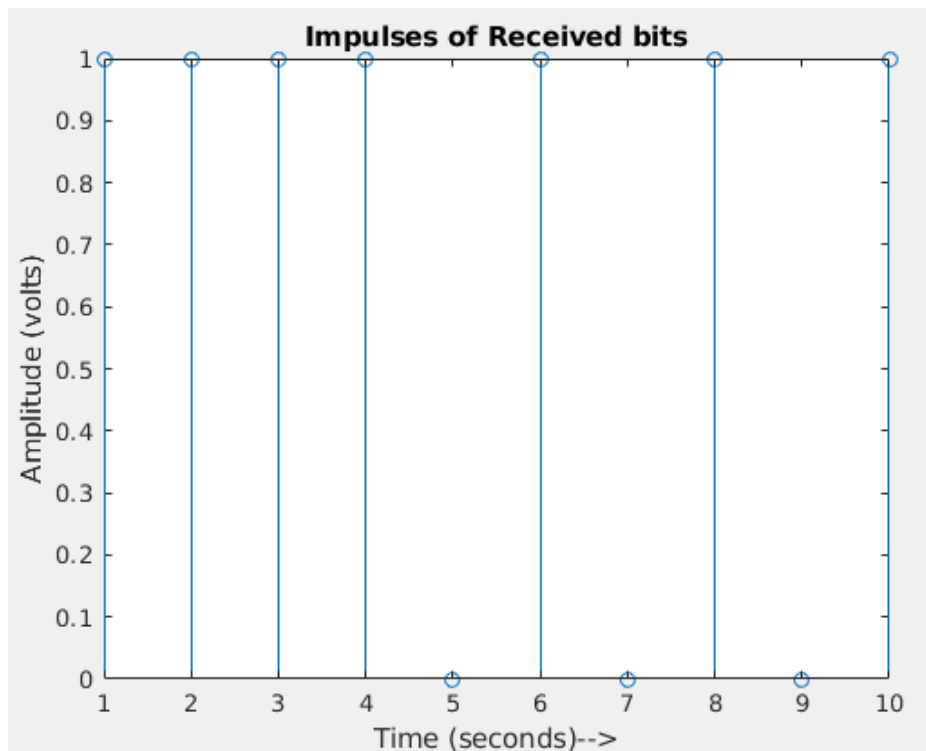


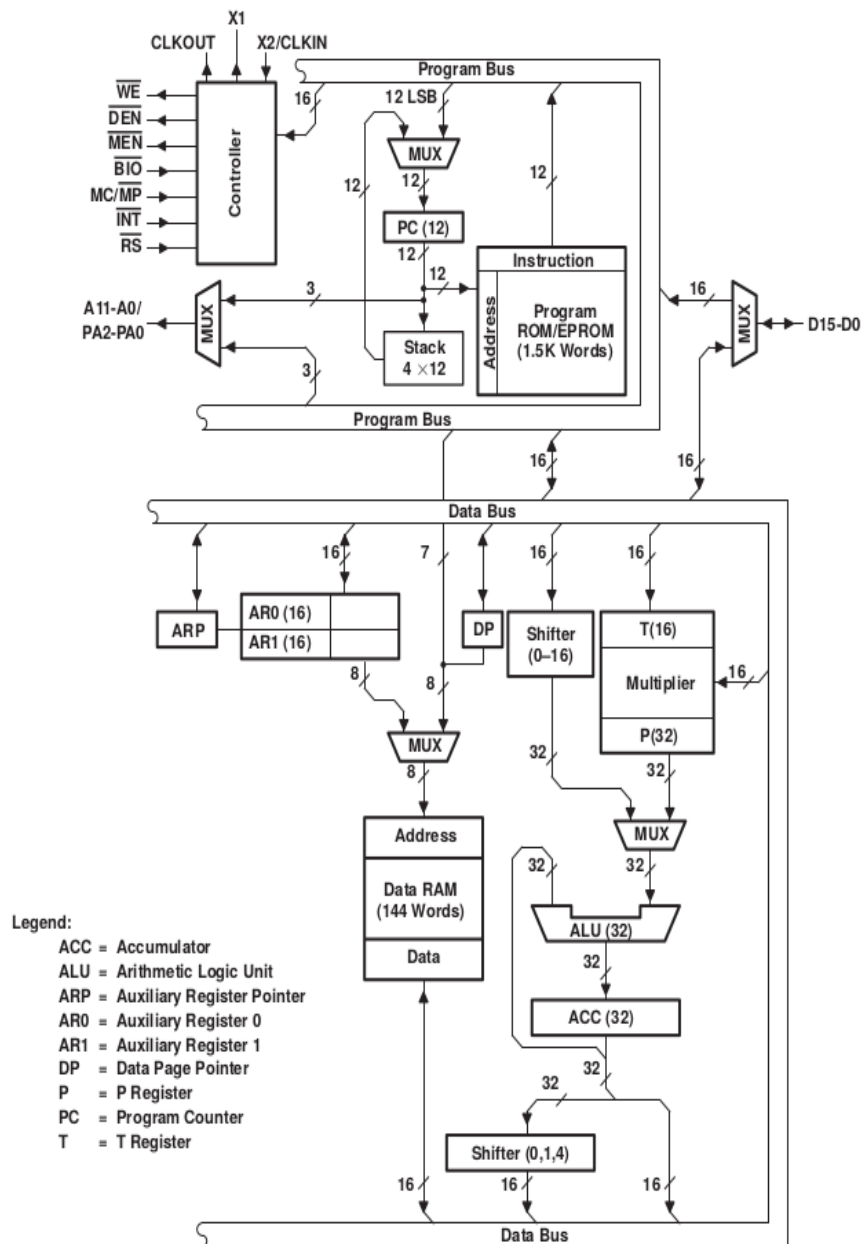
Рисунок 40 – Графік імпульсів отриманого інформаційного сигналу

Зчитуючи значення графіка, отримано наступну послідовність – $\{1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\}$, що дорівнює початковій послідовності, яку будуть передавати. Таким чином можна говорити про правильність побудованої моделі для приймача. Опорні сигнали в певній кількості випадків необхідно визначати емпіричним способом, в іншій можна використовувати так звані М-послідовності, які мають дуже гарні характеристики АКФ, або послідовності Баркера (коди Баркера). Все залежить від цілей, які необхідно досягти.

3.6 Синхронний приймач на основі DSP

Типова схема приймача вміщує в собі один із найважливіших компонентів для обробки сигналів – Digital signal processor. Для розробки синхронного приймача в рамках дипломної дисертації було обрано сигнальний процесор фірми Texas Instruments TMS320C15NL25. Даний сигнальний процесор має

достатньо сильні характеристики, що виділяють його з-поміж інших сигнальних процесорів за рахунок високої частоти ядра процесора, режимів роботи з плаваючою комою та можливість роботи з великими об'ємами оперативної пам'яті. На рисунку 3.24 зображено діаграму функціональних блоків даного процесора, а на рисунку 3.24 його зовнішній вигляді.





Підставою для вибору даного сигнального процесору є те, що його реалізовано на основі гарвардської архітектури, в якій основною перевагою є розділення шин доступу до вбудованої пам'яті та пам'яті даних. Завдяки такому рішення за один процесорний такт можна здійснити вибірку команди та даних. Також, даний процесор підтримує багатоядерне виконання, що особливо актуально для паралельного обрахунку значень ВКФ.

Зручність даного процесору полягає в тому, що він може бути працювати з широким переліком периферійних елементів: робота з оперативною пам'яттю DDR3, підтримка Enhanced Direct Memory Access, апаратна реалізація системи ФАПЧ, наявність співпроцесорів TCP3d (Turbo Decode Coprocessor) та VCP2 (Enhanced Viterbi-Decoder Coprocessor). Співпроцесори є дуже важливим критерієм вибору даного DSP. Далі наведено характеристики співпроцесорів як підстави для вибору даного процесору:

- TCP3d – швидкодіючий пристрій який прискорює каналне декодування для технологій WCDMA, HSPA, HSPA+, TD-SCDMA, LTE;
- VCP2 – співпроцесор для реалізації декодера Вітербі для мереж стандарту 3G, 2G.

Хоч дані співпроцесори є периферійними пристроями, проте вони з легкістю можуть інтегровані для роботи з DSP обраної марки. Це в свою чергу закладає фундамент для розробки приймача на основі сигнального процесору для сучасних технологій обміну даними.

Даний сигнальний процесор успішно може реалізовувати розроблену архітектуру синхронного приймача на основі ВКФ, що зображено на рисунку 3.24, де $y(t)$ – вхідний сигнал, $S_0(t)$, $S_1(t)$, $S_2(t)$, $S_N(t)$ набір опорних сигналів для знаходження значень ВКФ, r_1 , r_2 , r_3 , r_N відповідно значення ВКФ, а прямокутний блок відіграє роль елемента, що знаходить максимальне значення з-поміж усіх коефіцієнтів. Функціонально, основу синхронного приймача складає набір кореляторів (Correlator 1, Correlator N), які повинні обчислювати значення ВКФ паралельно, щоб визначити початок кожного кадру після отримання прапорця про

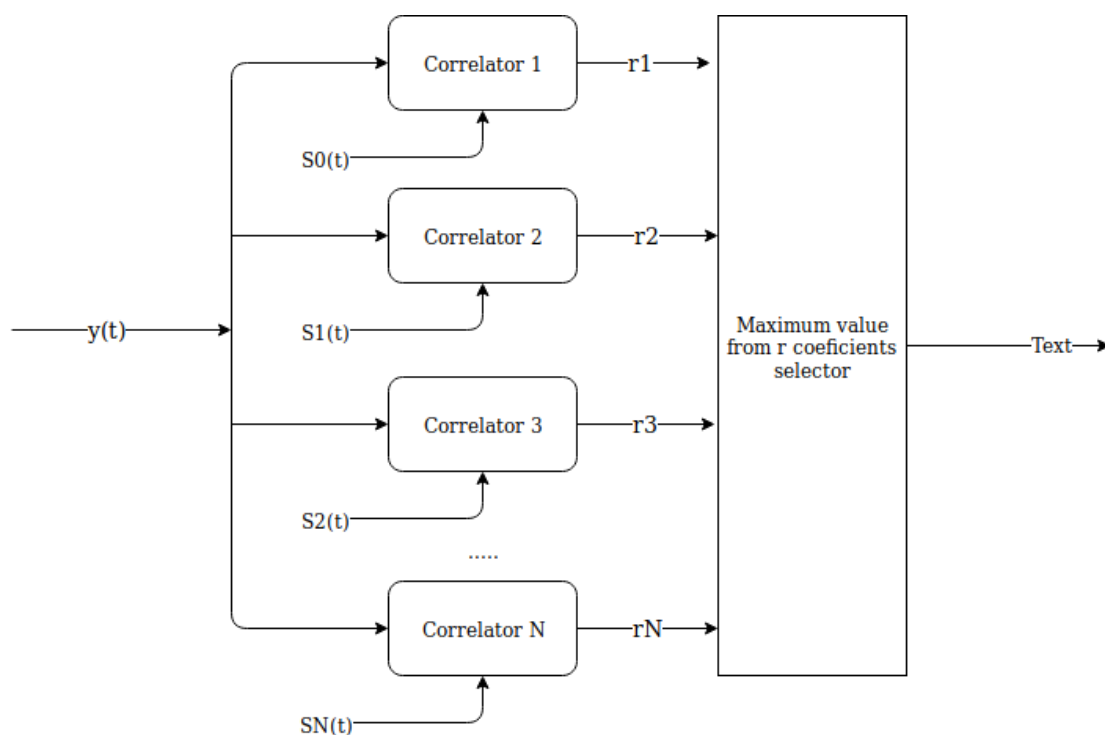


Рисунок 43 – Структурна схема кореляційного приймача

початок передачі, що відповідно впливає із синхронної природи приймача.

Схема кореляційного приймача, крім сигнального процесора, повинна містити наступні елементи: фільтри низьких частот, аналогово-цифровий перетворювач, цифро-аналоговий перетворювач, блок живлення, інтерфейс RJ45 для прийому сигналів, оперативна пам'ять (RAM), PROM. Оперативна пам'ять та EPROM необхідні для безпосереднього функціонування сигнального процесору.

Для схеми синхронного приймача слід використати EPROM у кількості двох штук для більш широких можливостей для програмування та роботи. Також, в схемі присутній аналогово цифровий перетворювач, щоб сигнальний процесора вже міг оброблювати виключно цифрові дані.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Вступ

Ідеї та напрацьована інформація, що викладені в даній магістерській дисертації мають практичну та комерційну цінність для області цифрової обробки сигналів. Цільове призначення програмного комплексу — розробка синхронізуючого програмного комплексу для роботи на спеціалізованій обчислювальній техніці. На даний момент існує декілька варіантів створення приймачів на основі DSP, FPGA, GPU та ASIC.

4.2 Маркетинговий аналіз стартап-проекту

4.2.1 Опис ідеї проекту

Опис ідеї проекту наведено в таблиці 2.

Таблиця 2 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Стартап проект передбачає створення зручного застосунку для роботи в області DSP, який дозволяє у зручній формі обирати різні методи для синхронізації сигналів, знаходження кореляції між наборами даних без написання специфічного коду для роботи	1. Навчальна сфера	Зручний графічний інтерфейс дає можливість наочно спостерігати процес синхронізації з використанням кореляції.

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	2. Синхронізація в системах передачі даних	Програмний застосунок може бути з легкістю адаптований для пристроїв цифрової обробки сигналів.
	3. Наочний розгляд графіків сигналів та кореляції	Можливість в режимі реального часу спостерігати на графіках залежності вхідних сигналів і кореляційних функцій, додавання шумів та емуляції реального синхронізуючого пристрої з боку приймача
		Можливість проводити симуляції процесів синхронізації та конфігурування параметрів середовища передачі даних, приймача тощо.

Зміст ідеї	Напрямки застосування	Вигоди для користувача
		Користувач не повинен володіти достатньо сильними навичками у математиці чи програмуванні для задання графіків функцій та інших параметрів програми.
		Застосунок можна використовувати як в UNIX подібних ОС так і в Windows ОС.

4.2.2 Аналіз потенційних техніко-економічних переваг

Аналіз сильних, слабких та нейтральних сторін стартап-проекту визначено в таблиці 5.2. Перевагами системи є наявність віддаленого доступу до ферм, підтримка великою кількістю майнінг-програм, зручний інтерфейс користувача, багатоплатформність, можливість моніторингу та наявність сповіщень.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Маг. проект	GPU	ASIC	FPGA			
1	Вартість	+	-	-	+			+
2	Високий рівень паралелізму	-	+	-	+	+		

3	Можливість програмування	+	+	-	+			+
4	Поширеність у сфері цифрової обробки сигналів	+	-	-	-			+
5	Малі габарити	+	-	+	+		+	
6	Простота реалізації мат. методів	+	+	+	-			+

Продовження таблиці 5.2

		(потенційні) товари/концепції конкурентів						
		Маг. проект	GPU	ASIC	FPGA			
6	Простота інтеграції з периферією	+	+	+	+		+	
7	Висока швидкість розрахунків	+	+	+	+			+
8	Можливість візуалізації процесів	+	+	-	+		+	

9	Розрахунок помилок в логарифмічній шкалі	+	-	-	-			+
10	Вбудовані елементи перетворення аналогових сигналів	+	+	-	+			+

4.3 Технологічний аудит ідеї проекту

Список технологій реалізації програмного комплексу наведено в таблиці 5.3.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технологій
1	Реалізація програмного забезпечення з використанням мови програмування Python, C/C++.	Мова програмування C/C++, Python та бібліотеки numpy, scipy, pyqtgraph	Технології активно розвивається в області open source	Знаходиться у відкритому доступі для користувачів
2	Створення кросплатформного графічного інтерфейсу з використанням мови програмування	Фреймворк pyqtgraph для реалізації кросплатформного графічного інтерфейсу на ОС Unix, Windows, Mac OS тощо.	Технологія доступна в open source	Доступна для розробників

	Python для візуалізації процесів			
3	Реалізація багатопотокового виконання для математичних обчислень	Використання модулю Thread мови Python	Технологія знаходиться у відкритому доступі	Доступна для розробників

Основна мета у виборі технологій — багатоплатформність, що і було досягнуто. Програмний комплекс потребує класичного набору інструментів та технологій, а саме: мови програмування Python для моделювання та C/C++ для реалізації, математичних бібліотек для роботи із числовими рядами, бібліотеки для багатопотоковості і розробки графічного інтерфейсу.

4.4 Аналіз ринкових можливостей запуску стартап-проекту

На даний момент, через відсутність великої кількості рішень, або явного конкурента, дане рішення готове до використання вже в мінімально робочій версії продукту. Даний програмний комплекс може повністю взяти нішу для майнінгу криптовалюти. Це можна побачити з даних наведених в таблиці 5.4.

4.4.1 Аналіз попиту

В таблиці 5.4 наведено дані з попереднього огляду ринку криптовалют. Умовною одиницею є проміжок часу, на який перемикаються майнери користувача для оплати користування застосунком. Це значення може варіюватись в залежності від навантаженості ферм і складає загалом від 5 хвилин до 20 хвилин.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	1000 грн/ум.од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Обмежень немає
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	80 %

4.4.2 Визначення потенційних груп клієнтів

Характеристики потенційних клієнтів стартап-проекту наведені в таблиці 5.5. Це можуть бути лише криптовалютні ентузіасти, або майнінг пули, які зацікавлені в повній автоматизації власних потужностей.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Вивчення галузі цифрової обробки сигналів	Навчальні заклади	Рівень попиту на процеси цифрової обробки сигналів може змінюватись	Просто графічного інтерфейсу та наочне відображення всіх характеристик та параметрів
Використання програмного	Компанії, що розробляють	Специфіка компаній, що	Швидка синхронізація сигналів з боку приймача

забезпечення для реалізації механізмів синхронізації в приймачах	пристрої цифрової обробки сигналів	розробляють процесори, може не потребувати даного програмного комплексу	незалежно від параметрів середовища передачі даних
--	------------------------------------	---	--

4.4.3 Аналіз ринкового середовища

Сфера криптовалют є відносно новою і зародилась на початку 2010-х років, тому у більшості країн не розроблена відповідна законодавча база для регуляції даної області, тому необхідно враховувати низку інших загрозливих факторів. Фактори ризику та загроз наведені в таблиці 5.6.

Таблиця 5.6 – Фактори загроз

№	Фактори	Зміст загрози	Можлива реакція компанії
1	Вихід на ринок нових конкурентів	З урахування помилок даного проекту можуть вийти рішення, які реалізують більшу кількість функціоналу та мають більше можливостей або працюють краще. Постійний розвиток проекту і введення додаткового функціоналу	Постійний розвиток проекту і введення додаткового функціоналу
2	Ріст цін на надпровідникові матеріали	Дешевша реалізація синхронізаторів на апаратному рівні ніж на програмному	Швидший випуск робочого продукту та здешевіння підтримки

3	Ліквідація навчальних установ в області систем передачі даних	Закриття навчальних закладів зменшує попит на потребу даного застосунок в області вивчення синхронізації сигналів	Збільшити відсоток використання продукту комерційними компаніями
---	---	---	--

Поточний розвиток галузі систем передачі даних не показує позитивних тенденцій з приводу розробки специфічних пристроїв чи програмного забезпечення, але при цьому потреба в наочних макетах, що можуть продемонструвати роботу приймачів в сучасних мережах є достатньо високою. Фактори можливостей наведені в таблиці 5.7.

Таблиця 5.7 – Фактори можливостей

№	Фактори	Зміст можливості	Можлива реакція компанії
1	Невелика кількість конкурентів	Відсутність монопольного ринку дає можливість та час для розвитку успішного продукту	Спостерігати за поточним станом ринку та реагувати на можливі їхні пропозиції у сфері DSP
2	Використання в спеціалізованих сферах (військова сфера)	Розвиток засобів РЕБ (радіоелектронної боротьби) потребує технічних та програмних засобів для розробки станцій радіолокації тощо	Запропонувати програмний комплекс для реалізації функцій РЕБ та навчання особового складу роботи з приймачами сигналів станцій
3	Відкриття нових технічних	Формування навчальної програми для	Необхідно слідкувати за станом наукової галузі та

	навчальних закладів	спеціальностей у сфері СПД	новинами про відкриття нових технічних навчальних закладів
--	---------------------	----------------------------	--

4.4.4 Аналіз пропозиції

Також, було розглянуто конкуренцію на ринку і результати наведені в таблиці 5.8.

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Тип конкуренції - чиста	Немає чітко виокремленого лідера серед конкурентів, та відповідно мала їх кількість	Сприятливо впливає на розвиток проекту
Рівень конкурентної боротьби - міжнародний	Сфера криптовалют зачіпає практично всі країни світу	Можливий швидкий вихід на світові ринки
За галузевою ознакою - внутрішньогалузева	Даний проект актуальний лише у сфері систем передачі даних	Слідкувати за трендами сфери криптовалют та відповідно розвивати потрібний функціонал
За видами товарів - товарно-родові	Задоволення потреб клієнтів за рахунок наявності максимальної спрощення	Тримати рівень спрощення не нижче поточного

	використання математичного апарату обробки сигналів	
--	---	--

4.4.5 Аналіз умов конкуренції в галузі

Аналіз конкуренції за М. Портером дало більш детально інформацію про стан ринку і можливості конкурентів.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Octave, Matlab, TI	Цінність ідеї та її якісна реалізація	Немає	Споживачі прямо впливають на успішність продукту, оскільки сплачують кошти за ліцензію користування	Конкуренти можуть стати дешевшими або стануть надавати якісніші послуги
Висновки:	Конкурентів небагато, відсутність	Є можливість виходу на ринок,	Немає постачальників	Так, реалізація потреб	Достатньо важко предентувати

	монополії, проте конкурент на боротьба значна за рахунок хорошої якості проектів конкуренті в	ідентичних продуктів немає і не передбачається		клієнтів вирішує успішність проекту	на клієнтів, які вже використовують інший готовий продукт
--	---	--	--	-------------------------------------	---

4.4.6 Перелік факторів конкурентоспроможності

Навіть при врахуванні конкурентів розроблена система має низку переваг, які наведені в таблиці 5.10.

Таблиця 5.10 — Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Розвинена галузь, але відсутні комплексі рішення для неї	Хоч сфера СПД в навчальних закладах достатньо широко розвинена теоретично, але практичних засобів демонстрації роботи синхронізаторів в приймачах немає і вивчення математичного апарату не дає повного уявлення роботи пристроїв, що приймають інформаційні сигнали. Також, багато цифрових пристроїв, що

		розробляються, для обробки цифрових сигналів також потребують програмного забезпечення для підтримки функціоналу синхронізації, оскільки інколи, апаратна реалізація може бути достатньо дорогою.
2	Ідея	Ідея створення комплексного рішення, яке демонструє роботу синхронізаторів, є унікальною, оскільки поточні рішення дозволяють проводити обрахунки лише після вивчення бібліотек застосунків.
3	Потреба в інвестиціях	Потреба в інвестиціях є, оскільки такого роду системи можуть вдосконалюватись та покращуватись довгий час за рахунок розробки нових стандартів синхронізації приймачів, які теж необхідно реалізовувати для демонстрації.
4	Швидкий вихід на ринок	Достатньо випустити мінімальну базову працездатну версію продукту на ринок для його використання
5	Команда	Інтелектуальний рівень команди розробників дає можливість якісно та швидко розроблювати необхідний функціонал. При цьому, важливим фактором є наявність технічної освіти, де вивчалась теорія систем передачі даних.
6	Інноваційність	Розробка специфічного програмного забезпечення з математичним ухилом, яке не вимагає дуже глибоких математичних знань.

4.4.7 Аналіз сильних та слабких сторін стартап-проекту

За наведеними факторами конкурентоспроможності в таблиці 5.10 було виконано порівняльний аналіз сильних та слабких сторін програмного комплексу (стартап-проекту) зі звязкою Awesome Miner, Miner Gate, який відображено в таблиці 5.11.

Таблиця 5.11 — Порівняльний аналіз сильних та слабких сторін системи віддаленого управління процесами та ресурсами майнінгових ферм

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні із системою управління процесами майнінгу						
			-3	-2	-1	0	+1	+2	+3
1	Розвинена галузь, але відсутні комплекси рішення для неї	12						+	
2	Ідея	19							+
3	Потреба в інвестиціях	7				+			
4	Швидкий вихід на ринок	20						+	
5	Команда	15					+		
6	Інноваційність	15							+

4.4.8 SWOT-аналіз

Складений SWOT-аналіз на основі вище отриманих даних відображено в таблиці 5.12.

Таблиця 5.12 — SWOT-аналіз стартап-проекту

Сильні сторони: — багатоплатформність;	Слабкі сторони:
---	-----------------

<ul style="list-style-type: none"> – широкий спектр функціоналу; – мала залежність від інвестицій; – невелика кількість сильних конкурентів; – зручний графічний інтерфейс користувача; – затребованість на ринку освіти; – всестороннє дослідження параметрів СПД та елементів кореляції; 	<ul style="list-style-type: none"> – потреба в оновленнях може бути достатньо низькою; – вузьке коло клієнтів;
<p>Можливості:</p> <ul style="list-style-type: none"> – невелика кількість конкурентів; – використання в спеціалізованих сферах (військова сфера); – відкриття нових технічних навчальних закладів; – інвестиції; 	<p>Загрози:</p> <ul style="list-style-type: none"> – вихід на ринок нових конкурентів; – ріст цін на надпровідникові матеріали; – ліквідація навчальних установ в області систем передачі даних;

4.4.9 Альтернативи ринкової поведінки

Визначення альтернатив було зроблено на основі часових термінів та ймовірності отримання ресурсів і відображено в таблиці 5.13

Таблиця 5.13 — Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтований комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації

1	Введення меншої вартості використання програмного комплексу	Ймовірно	2 місяць
2	Безкоштований пробний період	Дуже ймовірно	1 місяць

Найкраща альтернатива — безкоштовний пробний період для кожного користувача, який вирішить спробувати дану систему.

4.5 Розроблення ринкової стратегії проекту

Для розробки ринкової стратегії проекту необхідно спершу визначити стратегії охоплення ринку, а саме описати характеристики потенційних клієнтів стартап-проекту. Ця інформація наведена в таблиці 6.14.

4.5.1 Опис цільових груп потенційних клієнтів

В таблиці 5.14 наведено характеристики потенційних клієнтів.

Таблиця 5.14 - Характеристика потенційних клієнтів стартап проекту

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Навчальні заклади	Висока	85%	Низька	Просто
2	Військова сфера	Висока	75 %	Середня	Складно
3	Компанії, що розробляють пристрої обробки	Середня	45%	Висока	Складно

	цифрових сигналів				
Які цільові груп обрано: навчальні заклади					

За результатами аналізу потенційних клієнтів, було обрано дві цільові групи — майнінг пули та криптовалютні ентузіасти і на їх основі в таблиці 5.15 було обрано стратегію ринкового впливу.

4.5.2 Формування базових стратегій розвитку

Таблиця 5.15 — Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конку- рентоспроможні позиції	Базова стратегія розвитку
Пропонування програмного комплексу для технічних навчальних закладів	Диференційований маркетинг	Здатність пропонувати кращий та широкий спектр функціоналу	Стратегія диференціації

Базовою стратегією для стартап-проекту є стратегія диференціації, що означає надання програмному комплексу властивостей, які будуть відрізняти його від конкурентів.

4.5.3 Вибір стратегії конкурентної поведінки

В таблиці 5.16 наведено базові стратегії конкурентної поведінки.

Таблиця 5.16 — Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні	Залучати нових та забирати існуючих	Автомайнінг	Стратегія вилику лідера

Важливим елементом виходу на ринок є широкий спектр функцій, завдяки чому може відпасти потреба у використанні стороннього програмного забезпечення.

4.5.4 Розробка стратегії позиціонування

На основі потреб споживачів з обраних сегментів, а також в залежності від обраної базової стратегії розвитку в таблиці 5.17 наведено визначені стратегії позиціонування.

Таблиця 5.17 — Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного
-------------------------------------	---------------------------	--	---

			проекту (три ключових)
Максимальне спрощення використання математичного апарату Спостереження параметрів системи (середовище передачі, приймач) Синхронізація сигналів з ви- користанням кореляції	Стратегія диференціації	Розширення базового функціоналу	Автоматична синхронізація Приймач-емулятор Дослідження синхронізації сигналів в приймачах Вивчити синхронізацію сигналів в один клік

4.6 Розробка маркетингової програми стартап-проекту

Першим кроком до формування маркетингової концепції товару, який отримає споживач. Для цього у таблиці 5.18 підсумовано результати попереднього аналізу конкурентоспроможності товару.

4.6.1 Формування маркетингової концепції

В таблиці 5.18 визначено ключові переваги концепції потенційного товару

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Вивчення теорії синхронізації в приймачах	Програма в простій формі надасть необхідні кроки, які необхідні для запуску синхронізації тестових сигналів	Простота у користуванні (достатньо заповнити потрібні поля і натиснути декілька кнопок для запуску)
2	Відображення графіків сигналів і помилок	Відображення актуальних та статистичних даних у вигляді графіків	Швидка вибірка актуальних даних сигналів та інших статистичних вибірок
3	Реальна емуляція приймача	Обрати необхідні види сигналів, заповнити параметричні поля та натиснути одну кнопку	Проста емуляція приймача з користувацькими параметрами

На даному етапі необхідно розробити трирівневу маркетингову модель, де уточнюється ідея продукту, його фізичні складові, особливості процесу його надання. Ці дані наведено в таблиці 5.19.

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
Товар за задумом	Кореляційний синхронний приймача системи передачі даних

Рівні товару	Сутність та складові		
Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Дизайн 2. Високий рівень спрощення 3. Використання великої кількості математичних методів		
	Якість: стабільна робота ПЗ		
	Назва: Correlation Signal Synchronizator		
	До продажу: допомога із навчанням у користуванні програмою		
	Після продажу: реалізація додаткових функцій, підтримка, вирішення помилок		
За рахунок чого потенційний товар буде захищено від копіювання: обфускація програмного забезпечення та приватний доступ до кодової бази			

4.6.2 Визначення цінових меж

В даному розділі необхідно провести визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар. Визначена інформація вказана в таблиці 5.20.

Таблиця 5.20 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
--------------------------------	------------------------------	--	---

—	0 - 10000 грн	5000 грн	0 - 1500 грн
---	------------------	----------	--------------

4.6.3 Визначення оптимальної системи збуту

Необхідно визначити оптимальну систему збуту, в межах якого приймається рішення. Розрахунки наведено в таблиці 5.21.

Таблиця 5.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Покупка ліцензії	Створення проектної документації про програмний комплекс	Канал нульового рівня	Дистрибуція через Інтернет

4.6.4 Розробка концепції маркетингових комунікацій

Останньою складовою маркетингової програми є розробка концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів. Результати наведено в таблиці 5.22.

Таблиця 5.22 — Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
---------------------------------------	--	--	----------------------------------	--------------------------------

	цільові клієнти			
Ведення освітньої діяльності	Прямі офіційні	Послідовність в реалізації обраної позиції Доступність та об'єктивність інформації про фірму і товар Унікальність послуги Відповідність вимогам тендеру об'єктивність інформації про фірму і товар	Інформування цільової аудиторії про появу нового програмного рішення з акцентуванням на переваги Пояснення цільовій аудиторії відмінності та переваги в оплаті послуг	Раціоналістична стратегія реклами

4.7 Висновки

Під час проведення дослідження магістерської дисертації у якості стартап-проекту було створено відповідну аргументовану документацію. Протягом усього процесу дослідження було досягнуто повне розуміння цілей проекту, його переваги та недоліки, які в свою чергу дали можливість точніше визначити мету проекту та користувачку аудиторію програмного застосунку, визначити можливі методи монетизації, маркетингову стратегію для освітнього, військового та комерційного ринків.

Основними споживачами системи, що розроблена в рамках магістерської дисертації, є освітні заклади, військові замовлення та комерційні проекти.

На даний час попит на такий програмний комплекс може бути не дуже високим через специфіку області застосування.

Перевагами проекту є:

- просто у використанні;
- кросплатформність;
- зручний інтерфейс користувача;

Унікальність проекту полягає в рівні спрощення використання математичного апарату області DSP, спрощення графічного інтерфейсу для взаємодії та велика кількість відображень даних у вигляді графіків для наочного розуміння процесів, що відбувається в процесі синхронізації сигналів.

ВИСНОВКИ

Результатом дослідження та розробки в рамках магістерської дисертації є синтез кореляційного синхронного приймача системи передачі даних, який потенційно може бути використаний для прийому широкого спектру сигналів та їх перетворення. Було досліджено загальну проблематику при прийомі сигналів різного виду, їх синхронізація, визначення інформаційного сигналу з урахуванням завад, технічні рішення реалізації приймачів різного роду, що в свою чергу дало змогу краще зрозуміти об'єкт дослідження та якими характеристиками він повинен володіти.

Експериментальним шляхом було виявлено залежності для періодичних та неперіодичних сигналів з використанням автокореляції та взаємної кореляції, що було основою приймача в рамках магістерської дисертації. Було здійснено та проведено велику кількість моделювань засобами Matlab та Simulink для кращого розуміння сфери Digital Signal Processing.

Крім того, в магістерській дисертації було розглянуто декілька варіантів побудови синхронного приймача на основі технологій ASIC, GPU, DSP та ПЛІС. Кожен з розглянутих варіантів мав як свої переваги так і недоліки, що і розглянуто у відповідному розділі. Проаналізувавши кожен варіант реалізації, було прийнято рішення розробити комерційну модель кореляційного приймача для компанії Акронім Системс, що підтверджено відповідним актом впровадження, та окрема синтезована реалізація з використанням сигнального процесора, що є більш класичним підходом до побудови систем такого роду.

Також, в рамках магістерської дисертації розроблено стартап-проект, який має на меті вивести рішення на основі сигнального процесора на ринок мережевих пристроїв та любительського радіо. Розробка стартап-проекту допомогла краще досягнути рамки розроблюваного рішення, його недоліки та переваги, якими він повинен володіти. Це було закріплено у відповідних таблицях розділу.

Магістерська дисертація в повній мірі розкриває проблему синхронізації в приймачах, актуальність реалізації подібного рішення з використанням кореляційних процесів на основі сигнальних процесорів.

ПЕРЕЛІК ПОСИЛАНЬ

- [1] [Електронний ресурс] – 2018 . Режим доступу до ресурсу:
https://en.wikipedia.org/wiki/Digital_signal_processor
- [2] [Електронний ресурс] – 2018 . Режим доступу до ресурсу:
<https://www.analog.com/ru/design-center/landing-pages/001/beginners-guide-to-dsp.html>
- [3] [Електронний ресурс] – 2018 . Режим доступу до ресурсу:
<https://ru.wikipedia.org/wiki/RZ>
- [4] [Електронний ресурс] – 2018 . Режим доступу до ресурсу:
<https://ru.wikipedia.org/wiki/AMI>
- [5] [Електронний ресурс] – 2018 . Режим доступу до ресурсу:
<https://ru.wikipedia.org/wiki/NRZI>
- [6] [Електронний ресурс] – 2018 . Режим доступу до ресурсу:
https://ru.wikipedia.org/wiki/Манчестерское_кодирование
- [7] [Електронний ресурс] – 2018 . Режим доступу до ресурсу:
https://ru.wikipedia.org/wiki/Потенциальный_код_2B1Q
- [8] [Електронний ресурс] – 1991 . Режим доступу до ресурсу:
<https://www.digchip.com/datasheets/parts/datasheet/477/TMS320C15NL-25-pdf.php>
- [9] [Електронний ресурс] – 2011 . Режим доступу до ресурсу:
<http://www.ti.com/lit/ug/sprugs0/sprugs0.pdf>
- [10] [Електронний ресурс] – 2011 . Режим доступу до ресурсу:
<http://www.ti.com/lit/ug/sprugv6a/sprugv6a.pdf>
- [11] [Електронний ресурс] – 2018 . Режим доступу до ресурсу:
<https://allaboutfpga.com/fpga-architecture/>
- [12] [Електронний ресурс] – 2013 . Режим доступу до ресурсу:

<https://cyberleninka.ru/article/v/analiz-raboty-modeli-tsifrovogo-korrelyatsionnogo-priemnika>

- [13][
Електронний ресурс] – 1999 . Режим доступу до ресурсу:
https://www.mathworks.com/help/signal/correlation-and-convolution.html?s_tid=CRUX_lftnav
- [14]Б
аскаков С.І. Радіотехнічні лацнюги та сигнали, Вища школа, 2-е видання /
Баскаков С.І. - Москва, Вища школа, 1988. – 68 с.
- [15]С
кляр Б. Цифровий зв'язок. Теоретичні основи та практичне використання,
2-е видання / Скляр Б. - Київ, 2007. – 619 с.